DECODABILITY IS NOT TRANSFERABILITY: UNCOVERING LIMITATIONS IN GRAPH ATTRIBUTE AUDITING VIA ORTHOGONAL ALIGNMENT AND INVERTIBLE ROTATION CONTROLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Attribute inference auditing evaluates whether sensitive attributes beyond the intended task can be decoded from model outputs or internal representations - an important privacy concern for large language models that process graphs. Existing audits use output confidence or train attribute classifiers on shadow-model representations, assessing target-side attribute decodability through transfer to target representations. However, whether cross-model transfer reliably reflects target-side attribute decodability remains unclear. We systematically investigate this relationship through orthogonal alignment and invertible rotation techniques. Orthogonal alignment preserves attribute information while adjusting shadow representations' coordinate directions to better match target representations. Under identical preprocessing, alignment improves transfer performance by up to approximately 0.30 macro AUROC. This suggests that poor transfer may partly arise from coordinate mismatch and does not directly indicate that target attributes are difficult to decode. To further verify this, invertible rotation experiments change coordinate directions without removing information. After rotation, a linear classifier retrained on target representations decodes attributes with nearly unchanged performance, while the shadow-trained classifier's transfer performance changes. Thus, target-side decodability and cross-model transfer do not necessarily vary together. Overall, target-side decodability and cross-model transfer are not equivalent: low transfer scores do not directly imply that target attributes are difficult to decode. These findings reveal the limitations of assessing attribute exposure solely through transfer performance. Transfer-based audits should therefore interpret results in light of the classifier's training source and the characteristics of the representation spaces.

1 INTRODUCTION

Attribute inference attacks use model outputs, parameters, or internal representations to infer attributes of data that are not directly disclosed, including sensitive attributes beyond the model's intended task. These attacks can expose information that data holders wish to keep confidential even when the raw data are not released. Studies by Mehnaz et al. (2022), Wang & Wang (2022), and Zhang et al. (2022) show that these risks concern sensitive attributes of individual records, population-level properties of training graphs, and structural attributes of individual graphs. These findings indicate that withholding raw data is insufficient to prevent attribute leakage, making systematic auditing of model outputs and internal representations important.

As large language models are increasingly used for graph analysis, attribute leakage also warrants investigation in these systems. MolCA and LLaGA combine graph structure with language models' semantic understanding and text generation capabilities, creating new ways to analyze graphs and interact with them through natural language (Liu et al., 2023; Chen et al., 2024). Existing graph neural network studies use shadow models for attribute inference, providing a reference for privacy auditing in these systems (Wang & Wang, 2022; Yuan et al., 2024). In shadow-model auditing based on internal representations, the auditor obtains shadow representations from auxiliary data, trains a classifier using known attribute labels, and applies it to target representations to assess attribute

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

exposure through transfer performance. This procedure directly measures the cross-model transfer of the shadow-trained classifier. Target-side attribute decodability instead asks whether a classifier trained on target representations can decode the attributes. Whether these measurements agree requires evaluation. A central question is therefore whether a low transfer score indicates that target attributes are difficult to decode, or instead reflects limitations of the particular classifier, differences between the two representation spaces, and the information available to the auditor.

We compare target-trained and shadow-trained attribute classifiers across representation layers using two graph-classification datasets, two 8B language models, and five model seeds. On BACE, deepest-layer target-side decodability reaches 0.91–0.93 AUROC, while direct transfer is 0.45–0.58. Orthogonal alignment improves transfer by up to approximately 0.30 macro AUROC under identical preprocessing, supporting coordinate mismatch as one contributor to this gap. Invertible rotations further show that changing coordinates can alter transfer while leaving target-side linear decodability nearly unchanged. The gap persists under matched training and structure-based data splits.

Our main contributions are as follows:

- **Systematic evaluation of the gap between target-side decodability and cross-model transfer in graph–language models.** Layer-wise comparisons of target-trained and shadow-trained attribute classifiers show that low cross-model transfer scores can coexist with high target-side decodability, revealing a limitation of measuring target-side attribute exposure solely through cross-model transfer.
- **Controlled evidence that coordinate mismatch contributes to this gap.** Orthogonal alignment improves transfer under identical preprocessing, while invertible rotations change transfer while preserving representation information and linear decodability. Together, these results show that cross-model transfer scores depend on coordinate correspondence as well as whether the attributes are decodable.
- **Evaluation of the gap across conditions and recommendations for audit reporting.** Multiple model seeds, matched training, and structure-based splits assess the persistence of the gap in the tested settings. We recommend reporting target-side decodability and cross-model transfer separately, with the classifier's training source, available information, and uncertainty. When target-side evaluation is unavailable, it should be marked unassessed rather than replaced by a low transfer score.

2 Related Work

Attribute inference methods. Prior work infers attributes from model outputs or parameters, or directly from internal representations. Mehnaz et al. (2022) infer sensitive record attributes from confidence scores, while Wang & Wang (2022) use shadow models to infer population-level graph properties from target-model parameters or outputs. Representation-based methods train classifiers on labeled representations to recover sensitive text attributes (Song & Raghunathan, 2020) or structural graph attributes, such as size and density (Zhang et al., 2022). We focus on the classifier's training source: target-trained classifiers measure target-side attribute decodability, whereas shadow-trained classifiers evaluated on target representations measure cross-model transfer. We study the gap between these measurements and the effect of coordinate mismatch on transfer.

Evaluating attribute decodability. Classifiers are commonly used to examine which attributes can be decoded from representations. Their capacity, training procedure, and generalization affect the measurements, so poor performance by one classifier does not establish that the attribute cannot be decoded (Hewitt & Liang, 2019; Belinkov, 2022; Pimentel et al., 2020; Voita & Titov, 2020). Elazar & Goldberg (2018) show that newly trained classifiers can recover attributes even when the attribute classifier used during adversarial training performs poorly. These studies motivate distinguishing a particular classifier's failure from difficulty in decoding the attribute itself. We examine this distinction in cross-model auditing by comparing target-trained and shadow-trained classifiers and asking whether their different training sources lead to different assessments of attribute exposure.

Cross-model transfer and representation alignment. Domain-adaptation research identifies the effect of distribution differences on transfer performance (Ben-David et al., 2010) and improves

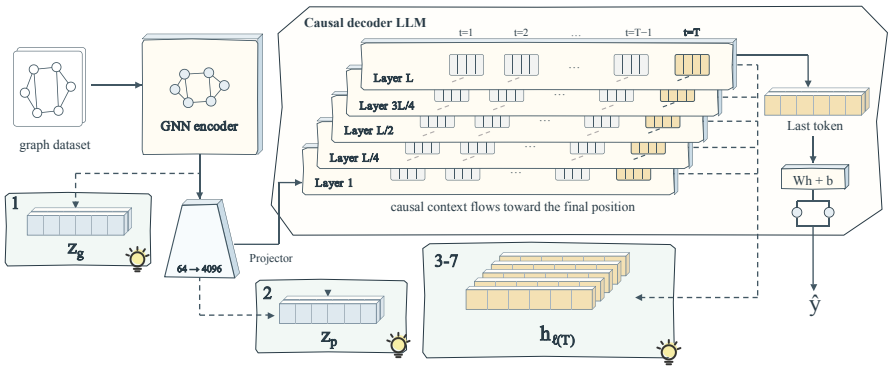


Figure 1: Graph-to-decoder computation and layer-wise representation extraction. Seven representations are extracted: the graph encoder output (z_g), the projector output (z_p), and five decoder hidden states at the final graph-token position, sampled at layers 1, $L/4$, $L/2$, $3L/4$, and L. Here, L denotes the total number of decoder layers: 32 for Llama-3.1-8B and 36 for S1-Base-8B. The final-layer state, after terminal normalization, also feeds the task head. Exact layer indices are listed in Section 4.1.

compatibility through representation alignment (Fernando et al., 2013). In privacy research, Champion et al. (2021) use Procrustes and Wasserstein–Procrustes alignment to recover some speaker identities after voice anonymization, demonstrating the importance of correspondence between representation spaces. We study this issue in graph representations, distinguishing target-side attribute decodability from cross-model transfer. Orthogonal Procrustes alignment (Schönemann, 1966) tests whether coordinate alignment improves transfer, while invertible rotations test whether transfer performance can still change when information and optimal linear decodability are preserved.

3 Methods for Attribute Auditing and Representation Analysis

We compare target-side attribute decodability with cross-model transfer and use alignment and rotation controls to examine their gap.

3.1 Graph Representations and Audit Interfaces

A graph encoder converts an input graph G into a representation $z_g = f_G(G)$. A projector then maps this representation to a continuous graph token $z_p = f_P(z_g)$, which is passed to the language model together with a text prompt. The decoder processes this token through its successive layers (Figure 1).

We extract representations at seven points: the graph encoder output, the projector output, and the graph-token hidden states at five decoder depths:

$$\mathcal{Z} = \{z_g, z_p, h_{\ell_1}, h_{\ell_2}, h_{\ell_3}, h_{\ell_4}, h_L\}. \tag{1}$$

Each point provides one representation per graph. We train a separate attribute classifier on the representations from each point to predict graph attributes. Representations from different points are evaluated separately and are not concatenated (Figure 1). Implementation details and the selected decoder layers are provided in Section 4.1 and Appendix A.

The target model is the model we audit. The shadow model produces representations for training attribute classifiers, which are then used to predict graph attributes from target representations. The two models use non-overlapping graphs for task training, and their graph encoders and projectors are trained separately. After training, we pass the same graph through both models and compare their representations. This step extracts representations from two already trained models; it does not train either model.

3.2 Attribute Decodability and Cross-Model Transfer

Using representations from the same observation point, we compare three evaluation paths: $T \to T$, $S \to T$, and $S \to S$. The model before the arrow produces the representations used to train the

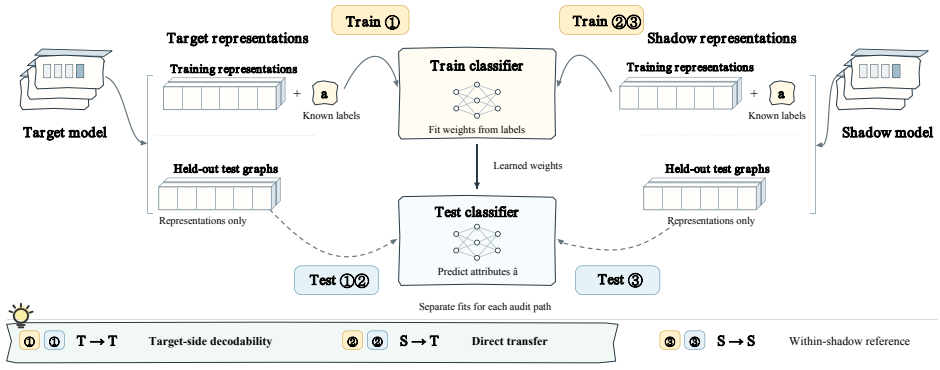


Figure 2: Classifier training and testing in the three audit paths. Target representations train the $T \to T$ classifier; shadow representations train the $S \to T$ and $S \to S$ classifiers separately. The first two paths share target test representations, and the third uses shadow test representations. Known labels train the classifiers; held-out labels are used only to calculate evaluation scores.

attribute classifier, and the model after the arrow produces those used to test it. We evaluate LR and MLP classifiers separately through all three paths, using the same graph attributes and labeling rules (Figure 2). Attribute definitions and classifier training details are provided in Appendices B and A.3.

$$P_{r\to s,k} = \mathrm{AUROC}\big(\hat{q}_{r\to s,k}(Z_s^{\mathrm{test}}), a_{s,k}^{\mathrm{test}}\big), \quad (r,s) \in \{(T,T),(S,T),(S,S)\}. \tag{2}$$

Here, $\hat{q}_{r\to s,k}$ is trained on representations produced by model r and their attribute labels. Z_s^{test} contains representations produced by model s for held-out test graphs, and $a_{s,k}^{\mathrm{test}}$ contains their true labels for attribute k. These test graphs are not used to train the attribute classifier, and their labels are used only to calculate AUROC, not to train or select classifiers.

Our main comparison is between target-side decodability, $D_k = P_{T\to T,k}$, and direct transfer, $A_k = P_{S\to T,k}$, as highlighted in Figure 2. These paths use the same target test representations and attribute labels, but train classifiers on target and shadow representations, respectively. The within-shadow score $W_k = P_{S\to S,k}$ provides an auxiliary reference. For $S \to T$ and $S \to S$, MLP classifiers use the same shadow training representations, attribute labels, architecture, and training procedure, but are initialized independently and tested on target and shadow representations, respectively (Appendix A.3).

Averaging equally across attributes gives D, A, and W. Our main contrast, $g = D - A$, measures the performance gap on the same target test set; positive values favor the target-trained classifier. These scores reflect performance under the specified training procedures, not representation information content; poor performance does not establish that other classifiers cannot decode the attributes.

3.3 Orthogonal Alignment

We obtain paired representations by passing the same validation graphs through the target and shadow models. We calculate shared standardization parameters and a PCA transformation ϕ from these paired representations, then apply the fixed transformation to the training, validation, and test representations used by the attribute classifiers. No attribute labels are used to compute the preprocessing or alignment. Let μ_S, μ_T be the preprocessed validation means, and let X_S, X_T be the centered matrices with corresponding rows from the same graph. Orthogonal Procrustes alignment (Schönemann, 1966) solves

$$Q^* = \arg\min_{Q^\top Q = I_d} \|X_S Q - X_T\|_F^2. \tag{3}$$

The shadow representations are transformed as

$$\widetilde{z}_S = (\phi(z_S) - \mu_S)Q^* + \mu_T. \tag{4}$$

Target representations undergo only standardization and PCA, while shadow representations undergo the same preprocessing followed by orthogonal alignment. To compare transfer with and without

alignment, both settings use the same paired validation graphs and preprocessing. Thus, $R = A_{\text{aligned}} - A_{\text{unaligned}}$ measures the change in transfer AUROC after adding alignment. The alignment step can be reversed and does not remove information from the PCA representations, although the preceding PCA reduction may lose information. Section 4.3 reports the number of validation graphs used and the experimental settings; Appendix C explains how the alignment matrix is computed and the classifiers are trained.

3.4 Invertible Rotations

We apply invertible rotations to the representations after PCA. These rotations change representation directions without removing information, because the original representations can be recovered by reversing the rotation. We use rotation matrices satisfying $Q^\top Q = I$. For a representation z, a linear classifier computes $s(z) = zw + b$. After rotating the representation to $z' = zQ$, we can adjust the classifier weights to $w' = Q^\top w$ to keep the score unchanged:

$$s'(z') = (zQ)(Q^\top w) + b = zw + b = s(z). \tag{5}$$

The unchanged scores give the same ranking and AUROC. Thus, every linear classifier before rotation has an equivalent classifier after rotation, although retraining LR may produce small numerical differences. When shadow and target representations use the same rotation, this weight adjustment also preserves the classifier's predictions on target representations. When they use different rotations, target predictions may change, so transfer AUROC may increase or decrease even though the rotations preserve information. Section 4.4 describes the four rotation settings and their comparisons; Appendix C.3 provides the full proof.

4 Experimental Setup and Results

4.1 Common Experimental Setup

Data and attributes. We use NCI1 (4,110 graphs) and MoleculeNet BACE (1,513 molecules), each with a binary graph-classification task (Morris et al., 2020; Wu et al., 2018). The five audited structural attributes are node count, edge count, density, diameter, and radius; diameter and radius use the largest connected component. Labels follow the training-derived threshold rule in Appendix B (Equation 9). BACE additionally provides molecular weight, AlogP, and ring count. These are input-derived benchmark attributes, not private annotations collected for this study.

Models and interfaces. Three Dense GraphSAGE layers (Hamilton et al., 2017) produce node representations that are concatenated, mapped to 64 dimensions, and mean-pooled over valid nodes. A linear projector and LayerNorm map the graph vector to a 4,096-dimensional token, appended to the native representations of the fixed prompt `Graph:`. The decoders are `ScienceOne-AI/S1-Base-8B` and `meta-llama/Llama-3.1-8B`, with 36 and 32 blocks, respectively (ScienceOne-AI, 2025; Dubey et al., 2024). The seven interfaces comprise z_g, z_p, and blocks (1, 9, 18, 27, 36) for S1 or (1, 8, 16, 24, 32) for Llama. The final graph-token state includes terminal normalization; a linear task head maps it to two task logits. Appendix A gives extraction details.

Training conditions and classifiers. We compare Frozen and LoRA settings to examine whether the relationship between target-side attribute decodability and cross-model transfer holds both with and without language-model fine-tuning. Within each model role and seed, Frozen and LoRA share graph-side weights, which remain fixed during decoder-stage training. Frozen trains only the task head; LoRA additionally trains low-rank decoder adapters (Hu et al., 2022). The main LoRA configuration uses rank 8, $\alpha = 32$, dropout 0.1, and $q/k/v/o$ projections in all decoder blocks. LR trains one classifier per attribute. The GELU-MLP uses hidden widths 256 and 128 and a separate output head per attribute. We use five model seeds (42–46) and five MLP classifier seeds (0–4); LR is deterministic. Full training and classifier settings appear in Appendices A.2 and A.3.

Evaluation and statistical conventions. Macro AUROC gives equal weight to the five structural attributes. Classifier-initialization scores are averaged within each model seed, and the five model seeds are the primary repetition units. Absolute-score tables and figure error bars report the mean and

sample standard deviation (SD) across model seeds. Paired changes use 1,000 paired hierarchical bootstrap replicates to obtain pointwise 95% confidence intervals. Alignment and rotation changes are reported with pointwise 95% confidence intervals, without multiple-comparison correction. Section 4.5 specifies the Holm and BH-FDR test families; resampling details appear in Appendices E and F.

4.2 Layer-wise Attribute Decodability and Cross-model Transfer

We examine whether low direct transfer corresponds to low target-side decodability. We pass the same audit graphs through the trained target and shadow models and extract representations at all seven observation points. The $T \to T$ and $S \to T$ paths use identical target test graphs. The graph encoder and projector were trained for 100 epochs at a learning rate of 10^{-3}. In the Frozen setting, only the task head was then trained for 10 epochs at 10^{-3}. In the LoRA setting, the adapters and task head were trained for 30 epochs at 5×10^{-4}. The target and shadow models used non-overlapping graphs for task training. Training and data-split details are provided in Appendices A.2 and B.1.

Table 1 reports the results at the final decoder layer. Across the two decoders, Frozen/LoRA settings, and classifier types, target-side decodability reaches 0.911–0.931 macro AUROC on BACE, while direct transfer reaches 0.449–0.581. On NCI1, the corresponding scores are 0.706–0.752 and 0.476–0.548. Within-shadow prediction also performs well, reaching 0.904–0.938 on BACE and 0.706–0.765 on NCI1. These results show that classifiers can predict attributes from both target and shadow representations when trained on representations from the same model, while shadow-trained classifiers perform poorly when applied to target representations.

Table 1: Final-decoder-layer macro AUROC, mean ± model-seed SD.

Setting		$\boldsymbol{D}$**:** $\boldsymbol{T \to T}$		$\boldsymbol{A}$**:** $\boldsymbol{S \to T}$		$\boldsymbol{W}$**:** $\boldsymbol{S \to S}$	
Decoder	**Condition**	**LR**	**MLP**	**LR**	**MLP**	**LR**	**MLP**
NCI1							
S1	Frozen	.706±.022	.737±.023	.523±.061	.506±.039	.708±.038	.728±.068
S1	LoRA	.714±.014	.707±.021	.513±.030	.509±.057	.708±.028	.706±.051
Llama	Frozen	.721±.016	.752±.010	.548±.034	.512±.047	.737±.027	.765±.011
Llama	LoRA	.710±.015	.723±.045	.492±.029	.476±.066	.711±.019	.728±.052
BACE							
S1	Frozen	.916±.020	.922±.013	.510±.095	.497±.068	.924±.017	.933±.016
S1	LoRA	.911±.046	.915±.041	.449±.105	.450±.081	.904±.032	.924±.024
Llama	Frozen	.927±.036	.931±.035	.487±.113	.503±.090	.930±.019	.938±.016
Llama	LoRA	.924±.029	.929±.036	.563±.060	.581±.110	.925±.026	.932±.029

Note. D, A, and W denote target-side decodability, direct transfer, and the within-shadow reference. MLP first averages classifier seeds within each model seed.

Figure 3 follows target-side decodability and cross-model transfer through all seven interfaces. Layer-wise changes describe how readily the chosen classifier decodes the attribute, and do not imply that deeper computation creates attribute information. The deepest layer is not necessarily the most decodable observation point in every setting. Appendix E.3 reports the descriptive maximum over sampled interfaces.

Additional experiments examine the relationships among the structural attributes and the signals used to predict them. Predicting each attribute from the other four yields macro AUROCs of 0.993 on NCI1 and 0.999 on BACE, showing that these attributes are closely related (Appendix B.3, Table 5). Classifiers trained on mean node features or representations from a GNN without task training also predict structural attributes, with the latter reaching 0.964 and 0.980 macro AUROC on NCI1 and BACE, respectively (Appendix B.3, Table 5). To examine the role of task labels, we predict attributes separately within each task class and rebuild the attribute labels and repeat the evaluation after subtracting each attribute's training-derived task-class mean. Target-side attribute prediction remains

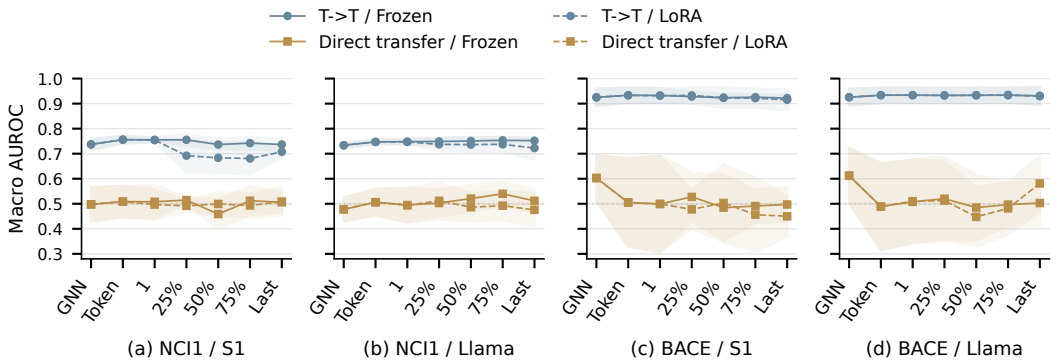


Figure 3: GELU-MLP results at seven interfaces. Blue circles show target-side decodability; ochre squares show direct transfer. Solid/dashed lines denote Frozen/LoRA. Shading is one standard deviation across five model seeds after averaging properties and classifier seeds. The light gray dotted horizontal line marks chance performance (AUROC 0.5); decoder depth is sparsely sampled.

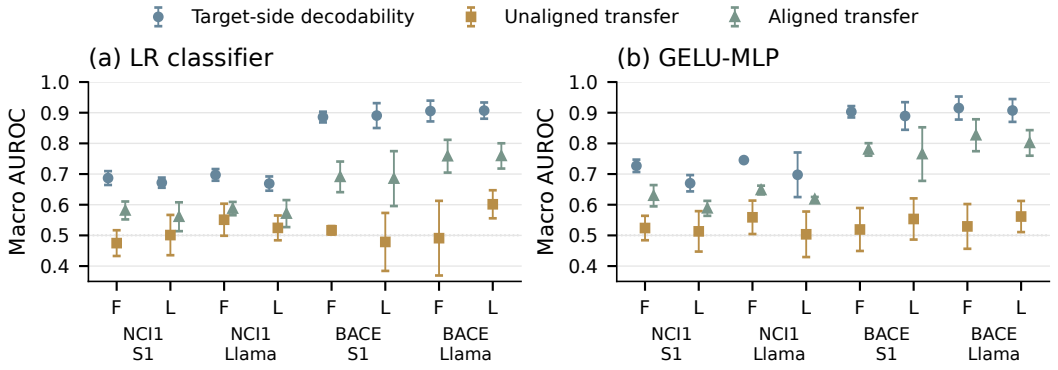


Figure 4: Target-side decodability, unaligned transfer, and transfer after orthogonal alignment in the same shared PCA-64 space. F/L denote Frozen/LoRA. Error bars show standard deviations across five model seeds. Aligned classifiers are retrained on transformed shadow representations.

effective under these controls (Appendix B.3). On BACE, molecular weight, ring count, and AlogP also show stronger target-side prediction than direct transfer (Appendix B.4, Figure 6).

These results show that attributes can be predicted from target representations even when classifiers trained on shadow representations transfer poorly to the target model.

4.3 Effects of Orthogonal Alignment on Transfer

We ask whether improving coordinate correspondence improves transfer. Each model seed provides paired validation representations of 247 NCI1 graphs or 91 BACE graphs, with each graph processed by both models. Both conditions use the same shared standardization and PCA-64 preprocessing from these paired validation representations; the aligned condition additionally transforms the shadow representations as described in Section 3.3. The alignment is learned without attribute labels.

Orthogonal alignment improves MLP transfer in all eight settings (Figure 4). All eight 95% bootstrap intervals exclude zero (Appendix C, Table 6). The largest gain is approximately 0.30 macro AUROC on BACE/Llama/Frozen, with interval [0.228, 0.369]. For LR, six of the eight intervals lie entirely above zero, while the two NCI1/Llama intervals include zero.

After alignment, transfer AUROC remains approximately 0.08–0.21 below target-side decodability, with both measured in the same PCA-64 space. Alignment changes the directions of shadow representations without removing information from the PCA-64 space or updating the target model.

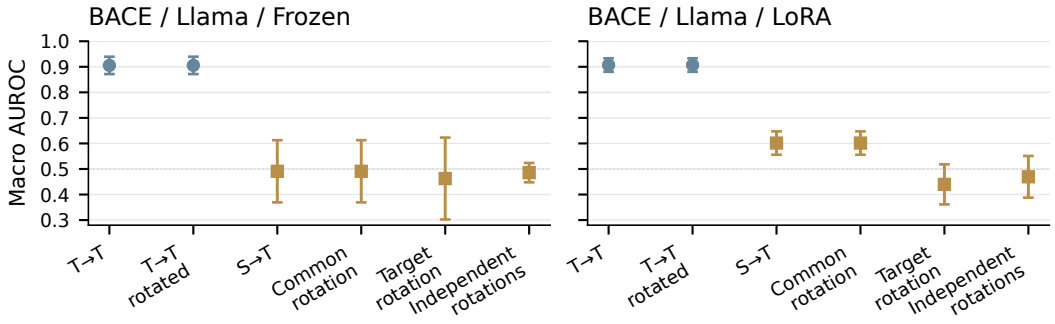


Figure 5: LR rotation controls on BACE/Llama in the fixed PCA-64 space, with Frozen on the left and LoRA on the right. Blue circles show target-side decodability before and after applying the same rotation to target training and test representations. Ochre squares show cross-model transfer without rotation, with a common rotation, with only the target rotated, and with independent shadow and target rotations. The gray dotted line marks chance performance (AUROC 0.5). Error bars show SD across five model seeds; paired-change confidence intervals are reported in Appendix C.2, Table 7.

Orthogonal alignment improves transfer performance, suggesting that differences in representation directions partly explain weak cross-model transfer.

4.4 Decodability and Transfer under Invertible Rotations

We ask whether changing coordinates can alter transfer while preserving representation information and linear decodability. PCA-64 preprocessing is held fixed, with no new coordinate-wise standardization after rotation. We evaluate four rotation settings based on Section 3.4: (i) $T \to T$: rotate target training, validation, and test representations with the same matrix and retrain LR, obtaining D_{rot}. (ii) $S \to T$: rotate shadow training/validation and target test representations with the same matrix and retrain LR on shadow representations, obtaining A_{common}. (iii) $S \to T$: rotate only target test representations and keep the original shadow-trained LR unchanged, obtaining $A_{\text{target only}}$. (iv) $S \to T$: rotate shadow training/validation and target test representations with different matrices and retrain LR on shadow representations, obtaining $A_{\text{independent}}$. We compare D_{rot} with the unrotated score D, and A_{common} with the unrotated score A. For the other two settings, we measure changes relative to the common-rotation control as $\Delta A_{\text{target}} = A_{\text{target only}} - A_{\text{common}}$ and $\Delta A_{\text{ind}} = A_{\text{independent}} - A_{\text{common}}$.

Consistent rotations leave the measured LR results nearly unchanged (Figure 5; Appendix C.2, Table 7). When target training and test representations receive the same rotation and LR is retrained, the maximum absolute mean change in target-side linear decodability is 2.87×10^{-6}. When both model roles receive a common rotation and the shadow LR is retrained, the corresponding maximum absolute mean transfer change is 1.36×10^{-5}. The measured differences are close to numerical precision, consistent with the score-preservation argument in Section 3.4.

Inconsistent rotations can instead alter transfer. On BACE/Llama/LoRA, rotating only the target changes transfer by -0.163, with 95% interval $[-0.239, -0.079]$, relative to the common-rotation control. Independent rotations change transfer by -0.132 relative to the same control (Appendix C.2, Table 7). Rotation effects depend on the setting; Figure 5 also shows the Frozen results for comparison.

Within the fixed PCA-64 space, invertible rotations preserve representation information and optimal linear decodability. When shadow and target representations undergo different rotations, however, the shadow-trained classifier can produce different scores and rankings on target representations, changing transfer AUROC (Appendix C.3). Alignment improves transfer by adjusting shadow representations to better match target representations. The rotation experiments further show that changing representation directions alone can alter transfer while leaving measured target-side linear decodability nearly unchanged. Together, these results show that a decrease in transfer performance does not necessarily indicate a decrease in target-side decodability. Supplementary representation-

similarity (CKA) results and experiments that classify whether a representation comes from the shadow or target model are provided in Appendix I.4.

4.5 Validation under Matched Training and Structure-based Splits

We ask whether the decodability–transfer gap persists after controlling training differences or changing the data partition. We analyze these two studies separately. Appendix I.5 summarizes their three final-layer MLP paths and transfer contrasts; complete results and settings appear in Appendices D and F.

Matched training: setup and results. To control for differences in training settings, Frozen and LoRA use the same graph-side and task-head initialization, optimization settings, training duration, and validation-based model selection within each target or shadow model and seed. The graph encoder and projector remain fixed; Frozen trains only the task head, while LoRA additionally trains adapters. We repeat $T \to T$, $S \to T$, and $S \to S$, with full training settings in Appendix A.2.

At the final decoder layer, MLP target-side decodability is 0.647–0.756 on NCI1 and 0.854–0.934 on BACE, compared with direct transfer of 0.462–0.515 and 0.484–0.538, respectively (Table 27). The gap therefore persists under matched training settings. MLP LoRA–Frozen transfer differences range from -0.050 to $+0.034$, with all four 95% confidence intervals including zero (Table 28).

Structure-based splits: setup and results. We group BACE molecules by molecular scaffold and NCI1 graphs by Weisfeiler–Lehman hash, keeping each group within one target or shadow training, validation, or test partition. We retrain the graph encoder and projector and repeat all three final-layer evaluation paths. The protocol also changes task-head initialization and learning rate; full settings and validation-based model selection are described in Appendix F.

On BACE/Llama, MLP target-side decodability is 0.930 under both Frozen and LoRA, while direct transfer is 0.530 and 0.399. On BACE/S1, the corresponding scores are 0.903/0.909 and 0.642/0.641 (Table 27). The gap persists under structure-based splits. All four MLP LoRA–Frozen transfer intervals include zero (Table 28). Appendix I.5 reports the individual contrasts and within-shadow scores.

Statistical interpretation. Within each study, Holm correction covers eight final-layer Frozen–LoRA transfer comparisons across four dataset–decoder settings and two classifier types; BH-FDR at 0.05 covers the exploratory comparisons (Holm, 1979; Benjamini & Hochberg, 1995). Neither study has a primary comparison passing Holm correction or an exploratory comparison passing BH-FDR at 0.05. These tests assess Frozen–LoRA transfer differences, while $D - A$ is reported descriptively. The evidence is insufficient to confirm an adaptation effect and does not establish equivalence. Full test definitions and supplementary analyses appear in Appendices E and F.

Taken together, the decodability–transfer gap persists across the tested training and split conditions. The structure-based study evaluates a changed training and partitioning protocol, so its differences from the matched study cannot be assigned to partitioning alone.

5 Conclusion and Future Work

This study reveals a limitation of shadow-model attribute auditing: weak cross-model transfer can coexist with strong target-side attribute decodability. Orthogonal alignment improves transfer, while invertible rotations can change transfer while preserving representation information and target-side linear decodability. Together, these results identify coordinate mismatch as one contributor to the decodability–transfer gap. The gap also persists under the tested matched-training and structure-based split protocols. These findings support reporting target-side decodability and cross-model transfer separately, with the classifier's training source and available information clearly stated.

Future work will extend this analysis to multiple graph tokens, generative graph–language tasks, and additional attribute types. We will also examine how data partitioning, model training, classifier choice, and the number of paired validation graphs affect the gap and the gains from alignment.

AI USE STATEMENT

Generative AI tools assisted with tasks requiring disclosure: Chinese-to-English translation and interpretation of the reported audit results, including checking whether textual summaries agree with saved experimental records; and research execution, where the research direction and central ideas were conceived by the authors and these tools assisted with writing parts of the analysis code (statistical tests, bootstrap procedures, and representation-extraction scripts) and with experiment execution and data processing. For tasks recommended for disclosure, these tools assisted with language editing, LaTeX formatting, figure-layout code, literature lookup, and identifying relevant references. Checks during manuscript preparation included comparing numerical summaries with saved result tables, recomputing reported statistics, replaying available saved predictions, and inspecting the compiled figures and cross-references. The authors are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

This work studies the limitations of attribute-inference auditing at graph–language model interfaces. The experiments use the public NCI1 and MoleculeNet BACE benchmarks and evaluate graph-level structural statistics and molecular descriptors. The study does not recruit human participants or collect private user data. The results describe predictive performance and uncertainty on benchmark graphs, rather than disclosures about identified individuals. Attribute-inference methods have dual uses: improving coordinate alignment can also strengthen cross-model inference. We therefore interpret the findings as guidance for evaluating and reporting attribute exposure, emphasizing that low transfer performance alone does not establish that an attribute is private.

REPRODUCIBILITY STATEMENT

The audit interfaces and evaluation paths are defined in Section 3; implementation and training settings are provided in Section 4.1 and Appendix A. Dataset membership, attribute thresholds, and confound controls are documented in Appendix B; alignment, rotation controls, and formal statements in Appendix C; statistical procedures in Appendix E; and the group-aware protocol in Appendix F. Appendix G records the execution environment, compute accounting, implementation provenance, and limits of independent reproduction. Additional numerical results are reported in Appendices D and I.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding Intermediate Layers Using Linear Classifier Probes. In *ICLR Workshop*, 2017. URL https://arxiv.org/abs/1610.01644.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2884. URL https://arxiv.org/abs/2306.03819.

G. W. Bemis and M. A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 2010. doi: 10.1007/s10994-009-5152-4.

Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Pierre Champion, Thomas Thebaud, Gaël Le Lan, Anthony Larcher, and Denis Jouvet. On the invertibility of a voice privacy system using embedding alignment. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 191–197, 2021. doi: 10.1109/ASRU51503.2021.9688159. URL https://arxiv.org/abs/2110.05431.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large Language and Graph Assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 7809–7823, 2024. URL https://proceedings.mlr.press/v235/chen24bh.html.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.14314.

Abhimanyu Dubey et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24 (43):1–48, 2023. URL http://jmlr.org/papers/v24/22-0567.html.

Yanai Elazar and Yoav Goldberg. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 11–21. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1002. URL https://aclanthology.org/D18-1002/.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2960–2967, 2013. doi: 10.1109/ICCV.2013.368. URL https://openaccess.thecvf.com/content_iccv_2013/html/Fernando_Unsupervised_Visual_Domain_2013_ICCV_paper.html.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.02216.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1626–1639, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.122. URL https://aclanthology.org/2021.emnlp-main.122/.

Sture Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2005.00687.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3519–3529, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. MolCA: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15623–15638, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.966. URL https://aclanthology.org/2023.emnlp-main.966/.

Shagufta Mehnaz, Sayanton V. Dibbo, Ehsanul Kabir, Ninghui Li, and Elisa Bertino. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4579–4596. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/mehnaz.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy*, 2019. URL https://arxiv.org/abs/1805.04049.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1806.05759.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A Collection of Benchmark Datasets for Learning with Graphs. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020. URL https://arxiv.org/abs/2007.08663.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420/.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.05806.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647/.

Peter H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451.

ScienceOne-AI. S1-Base-8B: Scientific foundation model. Hugging Face model card, https://huggingface.co/ScienceOne-AI/S1-Base-8B, 2025. Apache License 2.0.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011. URL http://jmlr.org/papers/v12/shervashidze11a.html.

Congzheng Song and Ananth Raghunathan. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020. URL https://arxiv.org/abs/2004.00053.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14/.

Xiuling Wang and Wendy Hui Wang. Group property inference attacks against graph neural networks. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2871–2884. ACM, 2022. doi: 10.1145/3548606.3560662. URL https://doi.org/10.1145/3548606.3560662.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science*, 9:513–530, 2018. doi: 10.1039/C7SC02664A.

Hanyang Yuan, Jiarong Xu, Renhong Huang, Mingli Song, Chunping Wang, and Yang Yang. Can graph neural networks expose training data properties? an efficient risk assessment approach. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/806288e682d8a38c0bf21e37ab38af0a-Abstract-Conference.html.

Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference Attacks Against Graph Neural Networks. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4543–4560, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/zhang-zhikun.

A Architecture, Training, and Observation Details

This appendix gives implementation details for the audit interfaces in Section 3.1 and the common experimental settings in Section 4.1.

A.1 Graph-to-decoder computation

$$z_g = f_G(G), \qquad z_p = f_P(z_g). \tag{6}$$

Each Dense GraphSAGE layer forms a neighbor mean from the binary adjacency and combines it with a separately projected root representation. ReLU, BatchNorm, and node masks are used in the graph encoder. The three 64-dimensional layer outputs are concatenated and mapped to a 64-dimensional node representation, then mean-pooled over valid nodes to obtain z_g. A linear map from 64 to 4,096 dimensions followed by LayerNorm gives z_p. BACE supplies nine-dimensional atom features; bond attributes are not used. This is a graph-classification interface with one graph vector, rather than an atom-token or text serialization system.

Table 2: Observation identity.

Item	**S1**	**Llama**
Model	`ScienceOne-AI/S1-Base-8B`	`meta-llama/Llama-3.1-8B`
Decoder blocks	36	32
Hidden width	4,096	4,096
Sampled block indices	1, 9, 18, 27, 36	1, 8, 16, 24, 32
Input	Native tokenization of `Graph:`, then one soft token	
Stored graph-side states	$z_g \in \mathbb{R}^{64}$ and $z_p \in \mathbb{R}^{4096}$	
Extraction precision	bfloat16 decoder forward pass; float32 saved states	

Note. The native tokenizer and vocabulary of each decoder are retained.

Decoder calls use `AutoModel`, `output_hidden_states=True`, and `use_cache=False`. Extraction runs in evaluation mode. For a sampled index k, the saved vector is exactly `outputs.hidden_states[k][:,-1,:]`. Intermediate tuple entries are post-block residual states at the graph-token position; the final entry is the terminal-RMSNorm output returned by the model. The final layer therefore has a distinct normalization convention that must be retained in reproduction. Pre-final-normalization states and separate attention/MLP sublayer outputs were not saved. No natural-language response, generated token, or repeated adaptive query is part of the property audit.

A.2 Training stages and matched controls

The layer-wise graph-side stage trains the GNN, projector, normalization, and projector-space classifier for 100 epochs using AdamW at learning rate 10^{-3} and weight decay 10^{-5}. The graph side is then frozen. The layer-wise frozen-decoder head uses 10 epochs at 10^{-3}; the LoRA condition updates its adapters and classifier for 30 epochs at 5×10^{-4}, with validation macro-F1 selection. Those schedules motivate the separate matched comparison, rather than supporting a claim that its layer-wise difference is due only to decoder adaptation.

The matched comparisons and configuration sweep comprise 140 task-training runs. The matched study starts both conditions from identical role- and seed-specific Phase-1 graph-side weights and classifier state. AdamW uses learning rate 5×10^{-4}, weight decay 10^{-5}, 30 epochs, batch size 32, and gradient clipping 0.5. The task objective is cross-entropy. The deterministic batch order and validation-selection rule are shared. Evaluation uses the model from the epoch with the highest validation macro-F1. Both conditions allow 30 training epochs; validation may select different epochs. The records distinguish the total training steps from the steps at the selected epoch.

Frozen trains the linear 4,096-to-2 task head, with 8,194 trainable parameters, on stationary decoder features. LoRA trains the same head and the declared adapter weights. The GNN, projector, and projector normalization remain in evaluation mode during adaptation; their weights and saved anchor states are fixed. Comparing anchors is especially important with BatchNorm, for which changing running statistics could otherwise alter representations despite frozen parameters.

Table 3: NCI1/S1 configuration sweep.

Condition	**Rank**	$\boldsymbol{\alpha}$	**Dropout**	**Adapted blocks**	**Task labels**
Frozen head	0	–	–	0	true
LoRA, $q/k/v/o$	8	32	0.1	36	true
LoRA, q/v	8	32	0.1	36	true
LoRA, final quarter	8	32	0.1	9	true
LoRA, rank 4	4	16	0.1	36	true
LoRA, rank 16	16	64	0.1	36	true
LoRA, no dropout	8	32	0.0	36	true
LoRA, permuted task	8	32	0.1	36	permuted

Note. Each row uses five model seeds and both model roles. The first two conditions also run on the other three dataset/decoder settings.

The random-task control uses a deterministic permutation of training labels, preserving their prevalence. Structural-property labels retain their original meanings. This control tests sensitivity to the optimization signal; it is not a random weight perturbation or an exact task-preserving intervention. Full fine-tuning and alternative PEFT algorithms are outside the experiment matrix. Adapter modules (Houlsby et al., 2019) and quantized low-rank adaptation (Dettmers et al., 2023) illustrate distinct adaptation choices that would require separate comparisons.

A.3 PROPERTY CLASSIFIERS

Classifier scores and evaluation definitions. Let $\mathcal{H}$ denote the classifier family. LR produces $q(z) = \sigma(zw + b)$, with $\sigma(u) = 1/(1 + e^{-u})$; the MLP produces scores through nonlinear hidden layers. Each classifier is fitted on its corresponding training representations and attribute labels and evaluated on graphs not used for fitting. For independent positive and negative examples Z^+ and Z^-, the AUROC definition including ties is

$$\mathrm{AUROC}(q) = \Pr\big(q(Z^+) > q(Z^-)\big) + \tfrac{1}{2}\Pr\big(q(Z^+) = q(Z^-)\big). \tag{7}$$

Across K attributes, $\mathrm{macroAUC} = K^{-1}\sum_{k=1}^{K}\mathrm{AUROC}_k$. Macro averaging summarizes attributes without assuming their independence. The three attribute-level paths are defined in Section 3.2 (Equation 2). The reported scores average these attributes:

$$D = \frac{1}{K}\sum_k D_k, \qquad A = \frac{1}{K}\sum_k A_k, \qquad W = \frac{1}{K}\sum_k W_k. \tag{8}$$

The main-text gap is $g = D - A$. The paths differ in the training information available to the classifier, with D and A evaluated on the same target test graphs. As detailed below, the MLP fits have path-specific initializations.

LR operates on standardized representations and predicts each property separately. The corrected nonlinear classifier has the shared trunk

$$\begin{aligned}\mathrm{LayerNorm}_{\text{no affine}}(d) &\to \mathrm{Linear}(d, 256) \to \mathrm{GELU} \to \mathrm{Dropout}(0.1)\\ &\to \mathrm{Linear}(256, 128) \to \mathrm{GELU},\end{aligned}$$

followed by a separate two-logit head for each property. Its loss is the mean cross-entropy across property heads. AdamW uses learning rate 10^{-3} and weight decay 10^{-4}, batch size 256, and gradient clipping 1.0. Training lasts at most 100 epochs, with validation macro-AUROC saved model selection, improvement tolerance 10^{-5}, and a 15-epoch no-improvement stopping rule after the initial 21 epochs. Five classifier seeds (0–4) are paired across conditions. The reported LR is deterministic with classifier-seed index 0.

Raw classifiers use a standardizer fitted to the corresponding attack-training representations. PCA-64 diagnostics additionally fit a 64-component randomized PCA to those training states; the component count cannot exceed the available sample count or feature dimension. The alignment experiment is separately defined: its common standardizer and PCA are fitted using paired validation states, so it has a stronger calibration budget than a raw transferred classifier. The rotation diagnostic uses LR with `liblinear`, `max_iter=10000`, and `tol=1e-8` to control numerical convergence.

An earlier two-layer property head had no intervening activation and thus represented a deep linear model. It is not treated as nonlinear evidence here. The main text uses the corrected GELU-MLP. A separate layer-wise task classifier, Linear(d,256)–ReLU–Dropout(0.1)–Linear(256,2), estimates task-label accessibility; it must not be confused with the deployed end-to-end task head used for utility and noninferiority.

LR uses fixed hyperparameters rather than validation-based hyperparameter selection. The layer-wise attack script and the matched attack script reused by the group-aware replication include the path name in the deterministic MLP seed derivation. Consequently, $S \to S$ and $S \to T$ are separately fitted networks under the same architecture and training procedure, rather than two evaluations of an identical saved model. Within a path, the classifier-seed indices are paired across Frozen/LoRA and averaged within each model seed. The notation $\hat{q}_{r\to s,k}$ in Section 3.2 distinguishes these separately trained classifiers; the MLPs used for W and A have different initial parameters. Within-shadow performance is therefore a reference for the fitting procedure's ability to learn shadow attributes. A protocol that reuses one fitted shadow network on both test domains would further isolate the transfer comparison.

B DATA MEMBERSHIP, PROPERTY DEFINITIONS, AND CONFOUND CONTROLS

The definitions and controls below support Sections 3.2 and 4.2.

B.1 Layer-wise and common audit memberships

After graph-size filtering, NCI1 contains 4,110 graphs and BACE 1,513. The graph-size cap is 128 nodes. Original class counts are 2,053/2,057 for NCI1 and 822/691 for BACE; mean node/edge counts are approximately 29.9/32.3 and 34.1/36.9, respectively. Binary molecular adjacency omits bond types, so this interface is not a complete representation of chemical identity.

Target and shadow task-model memberships are separate. The layer-wise attack audit encodes the same locked graph IDs with both role models; a role label on a representation denotes the model that produced it, not a different input graph. The common attack train/validation/test counts are 986/247/411 for NCI1 and 362/91/154 for BACE. Where layer-wise validation indices were absent, the validation partition was taken from the original training membership with an 80/20 subdivision. This common-population diagnostic controls input identity when measuring coordinate compatibility. It is distinct from a fresh task-model holdout or a deployment in which the classifier draws an unrelated auxiliary population.

All classifier paths reuse property thresholds, graph IDs, row order, and test membership across Frozen/LoRA. Target test labels are used to score classifier outputs, not to fit the classifiers or the alignment. Layer-wise datasets and model artifacts were nevertheless examined during earlier development, and the matched rerun inherits those memberships. Thus the completed layer-wise and matched studies are evidence about those audited populations, not an untouched group-disjoint generalization test.

B.2 Thresholds and structural labels

For each continuous property $v_k(G)$, the binary target is

$$a_k(G) = \mathbf{1}\{v_k(G) \geq \tau_k\}. \tag{9}$$

Attribute definitions and thresholds remain fixed within each comparison; test data are not used to select thresholds. Thresholds use the inherited upper-median rule on locked attack-training graphs and are applied unchanged to validation and test. For zero-indexed sorted values $v_{(0)}, \ldots, v_{(n-1)}$, the implemented rule is

$$\tau = \tfrac{1}{2}\left(v_{(\lfloor n/2 \rfloor)} + v_{(\lceil n/2 \rceil)}\right).$$

It should be reproduced literally rather than silently replaced by a library's default median convention. Ties can produce a positive prevalence different from one half. Node count uses the valid-node mask, and edge count is half the number of nonzero entries in the masked binary adjacency. Density is $|E|/\binom{|V|}{2}$ (zero for at most one node). Diameter and radius are the maximum and minimum vertex eccentricities on the largest connected component, respectively. The retained continuous property values and graph IDs are the authoritative targets for reproducing these layer-wise attack results.

B.3 Input and task-signal baselines

The direct-statistic baseline includes the queried property and is a construction sanity check, not a nontrivial classifier. The leave-one-property-out baseline removes that statistic and predicts its binary label from the other four continuous statistics. Its near-perfect performance demonstrates redundancy. Node-feature means remove the trained graph encoder, while the untrained GNN retains the graph architecture without task learning. Task-logit baselines use the model's two task scores as features. All learned baseline preprocessing uses attack-training data.

Task-conditioned attacks fit and test a property classifier separately within each task class. Residual attacks first estimate $\mu_{k,y} = \mathbb{E}[v_k \mid y]$ on attack-training membership, then compute $r_k(G) = v_k(G) - \mu_{k,y(G)}$ and binarize with a training-derived threshold. Both analyses use known task labels as diagnostic information, and neither controls for every aspect of the task distribution. Conditional classifier-based evaluation (Hewitt et al., 2021) similarly distinguishes classifier output beyond a baseline; our within-class and residual-label controls use a different estimator and do not estimate conditional $\mathcal{V}$-information. Continuous residualization and subsequent thresholding are different from subtracting a fitted task classifier's confidence. We do not claim that task-label conditioning proves causal independence from intended-task features.

For NCI1/Llama Raw+LR, the residual target-side LoRA–Frozen difference is -0.011 with pointwise 95% interval $[-0.040, 0.012]$, while the residual transfer difference is -0.033 $[-0.063, -0.002]$.

Table 4: Layer-wise held-out property distributions.

Data	Property	Threshold range	Mean	Value range	Positive fraction
NCI1	Nodes	26–27	29.808	3–111	0.484–0.567
	Edges	29–29	32.274	2–119	0.499–0.538
	Density	0.0826–0.0855	0.089	0.0192–0.667	0.470–0.547
	Diameter	12–12	13.293	2–45	0.540–0.589
	Radius	6–6	6.964	1–23	0.672–0.710
BACE	Nodes	33–33	34.126	11–97	0.494–0.539
	Edges	36–36	36.903	11–101	0.474–0.526
	Density	0.0677–0.0695	0.070	0.0217–0.218	0.481–0.532
	Diameter	15–15	15.287	6–41	0.519–0.584
	Radius	8–8	7.901	3–21	0.519–0.584

Note. Threshold and prevalence ranges span the ten decoder/seed records per dataset. The mean is averaged over those records; the value range is the observed minimum to maximum. The ranges summarize the saved split-specific distributions and thresholds.

Table 5: Input and task-label controls.

Data	Property–task	Property–property	Leave-one-out	Node mean	Random GNN	Task logits
NCI1	.342	.993	.993	.637	.964	.572
BACE	.329	.997	.999	.890	.980	.656

Note. AUROC is macro-averaged across structural properties. Correlations are the largest absolute mean held-out Spearman correlation over the relevant pairs.

For GELU-MLP, the corresponding differences are -0.035 $[-0.076,-0.001]$ and $+0.001$ $[-0.030,0.025]$. Their different directions again show why a decodability contrast cannot substitute for a transfer contrast. These are exploratory layer-wise estimates.

Structural and task-label controls. The five coarse properties form a strongly dependent structural bundle. The largest absolute mean held-out property–property Spearman correlation is 0.993 on NCI1 and 0.997 on BACE. A leave-one-property-out LR predicts the omitted binary property at 0.993 and 0.999 mean AUROC. Consequently, macro averaging summarizes related measurements; it does not count five independent disclosures.

Input-side baselines identify an additional source of easy classifier output. An untrained mean-pool GNN attains 0.964/0.980 macro AUROC on NCI1/BACE, and node-feature means attain 0.637/0.890. This shows how much structural decodability is already available before task learning. Task logits are weaker at 0.572/0.656, while the largest absolute property–task correlation is 0.342/0.329. High representation AUROC cannot therefore be attributed entirely to a new decoder-induced privacy risk.

The signal is also not exhausted by the discrete task label. After subtracting the attack-training conditional property mean given the task label and rebinarizing the residual, target-side GELU-MLP AUROC is 0.716/0.683 for NCI1/S1 Frozen/LoRA, 0.895/0.895 for BACE/S1, and 0.914/0.910 for BACE/Llama. Within-task-class attacks provide a complementary check. These adjustments establish residual decodability relative to the observed label; they do not condition on every task-relevant molecular factor.

BACE descriptors show a similar access-path separation. Target-side LR AUROC is approximately 0.94–0.95 for molecular weight, 0.86–0.91 for ring count, and 0.72–0.77 for AlogP, whereas direct transfer is near chance (Appendix B.4). Their sensitivity depends on the application: they can be restricted molecular attributes when the authorized output is an activity label, but they are not private annotations in these public benchmarks.

B.4 MOLECULAR DESCRIPTORS

MW, AlogP, and ring count are read from the original BACE CSV after raw-ID and task-label alignment. The identity audit checks 607 common-population rows per seed/decoder record, exact task-label agreement, and finite descriptor values. The identity check records the SHA-256 hash of the BACE CSV. Descriptor thresholds are fitted on attack-training membership, and the same binary labels are used for both role representations.

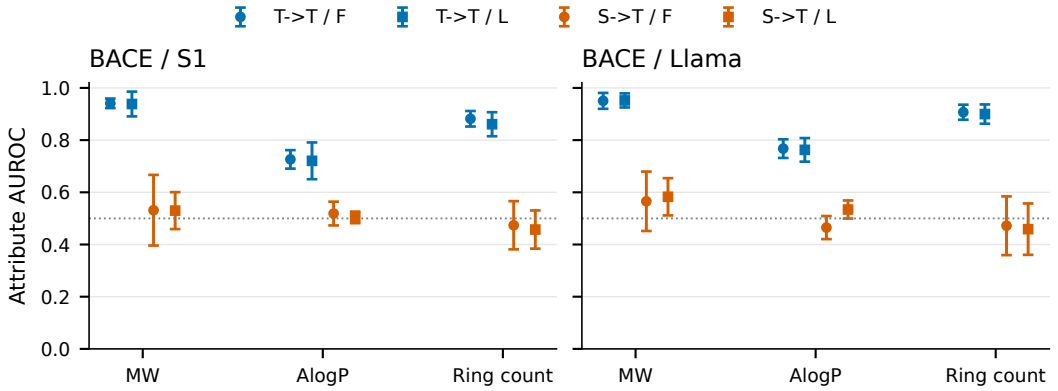


Figure 6: Layer-wise BACE chemical-descriptor LR attacks at the deepest sampled layer. F/L denote Frozen/LoRA. Color distinguishes target-trained decodability from transfer, while marker shape distinguishes branch. Error bars are one standard deviation across five model seeds. All AUROC axes span 0–1.

The molecular descriptors are not ground-truth private annotations collected for this study. They illustrate attributes that could fall outside the authorized output in a molecular service. Because they correlate with structure and with one another, successful descriptor decodability does not establish independent disclosure events. The descriptor analysis uses the same model-seed and classifier-seed aggregation as the structural-attribute analysis.

C ALIGNMENT, INVARIANCE, AND TRANSFER CONDITIONS

This appendix supplements the methods in Sections 3.3 and 3.4 and the experiments in Sections 4.3 and 4.4. All invertibility statements refer to the fixed projected space, not the preceding PCA reduction.

C.1 CALIBRATION PROTOCOL AND DIAGNOSTIC COMPARISON

Closed-form solution and score contrast. For centered paired matrices $X_S, X_T \in \mathbb{R}^{n\times d}$, the Frobenius norm $\|\cdot\|_F$ is the square root of the sum of squared matrix entries. The constraint $Q^\top Q = I_d$ preserves lengths and angles, allowing rotations and reflections. The singular value decomposition gives

$$X_S^\top X_T = U\Sigma V^\top, \qquad Q^* = UV^\top. \tag{10}$$

Both conditions retain the same preprocessing and paired-data budget. Their transfer contrast is

$$R = A_{\text{aligned}} - A_{\text{unaligned}}. \tag{11}$$

Each model seed uses 247 NCI1 or 91 BACE paired validation graphs. The same graph is evaluated by the shadow and target models. Their stacked validation representations fit one standardizer and one PCA basis. We center the two role matrices separately and solve Equation 3; aligned shadow training/validation vectors are $(\phi(z) - \mu_S)Q^* + \mu_T$, where ϕ is the shared standardization and PCA transformation. The classifier is refitted on transformed shadow training vectors. MLP saved model selection uses transformed shadow validation vectors and their attribute labels; LR retains fixed hyperparameters. All final classifier outputs are evaluated against the unchanged target-test vectors

in the shared PCA space. The experiment records retain preprocessing, pairing, and saved-model hashes for these comparisons.

Table 6: Layer-wise orthogonal-alignment recovery, aligned minus unaligned transfer in PCA-64.

Data	Decoder	Branch	Classifier	Recovery	Pointwise 95% CI
NCI1	S1	F	LR	+0.105	[0.059, 0.149]
		F	MLP	+0.105	[0.040, 0.159]
		L	LR	+0.060	[0.017, 0.103]
		L	MLP	+0.074	[0.027, 0.125]
BACE	S1	F	LR	+0.174	[0.126, 0.219]
		F	MLP	+0.260	[0.172, 0.345]
		L	LR	+0.205	[0.107, 0.340]
		L	MLP	+0.209	[0.123, 0.312]
NCI1	Llama	F	LR	+0.037	[-0.004, 0.074]
		F	MLP	+0.090	[0.047, 0.134]
		L	LR	+0.046	[-0.003, 0.090]
		L	MLP	+0.114	[0.051, 0.176]
BACE	Llama	F	LR	+0.265	[0.156, 0.384]
		F	MLP	+0.298	[0.228, 0.369]
		L	LR	+0.158	[0.104, 0.210]
		L	MLP	+0.240	[0.162, 0.319]

Note. Intervals are pointwise paired hierarchical 95% intervals. Values here are the bootstrap summary estimates; small differences from the plotted direct means reflect the recorded aggregation.

The unaligned PCA comparator already uses a basis fitted on both validation roles. Thus the incremental benefit in Table 6 is the benefit of alignment beyond shared unsupervised preprocessing. It should not be presented as the result of a zero-calibration attack. Target-side diagnostic training is also a stronger access path than ordinary shadow transfer; a high D establishes accessibility to that trained classifier, not success for every classifier.

C.2 Invertible rotations

Figure 5 in the main text reports both Frozen and LoRA. Preprocessing is fixed before any rotation. Orthogonal matrices are generated by QR decomposition of a Gaussian matrix with the signs of the diagonal of R used to fix the column signs. Consistent target rotations transform target training, validation, and test vectors together before LR is refitted. Common rotations transform shadow training/validation and target test vectors by the same matrix before the shadow classifier is refitted. Target-only rotations retain the shadow training coordinates and rotate the target test vectors. Independent rotations use separate matrices for the two roles. The paired changes for both inconsistent controls use the common-rotation condition as the reference.

Table 7 summarizes the rotation controls. The consistent-control rows report maximum absolute mean changes across the reported checks; the inconsistent-control rows report signed changes for BACE/Llama/LoRA relative to the common rotation. The small consistent-control changes quantify finite-precision training behavior, while Appendix C.3 establishes invariance of the optimal linear scoring family in the fixed space. PCA itself is not asserted to be lossless.

C.3 Formal statements

Population decodability and fitted classifiers. For a fixed target distribution and attribute k, optimal linear decodability is

$$D^{*,\mathrm{lin}}_{T,k} = \sup_{w,b} \mathrm{AUROC}(Z_T w + b, a_{T,k}). \tag{12}$$

Table 7: LR rotation controls in PCA-64.

Coordinate control	Measurement	AUROC change
Consistent rotations: maximum absolute mean change		
Consistent target rotation	Target-side linear decodability	2.87×10^{-6}
Common rotation of both roles	Transfer	1.36×10^{-5}
Inconsistent rotations: BACE/Llama/LoRA signed mean [Pointwise 95% CI]		
Target-only rotation	Transfer	-0.163 $[-0.239, -0.079]$
Independent role rotations	Transfer	-0.132 $[-0.180, -0.087]$

Note. Consistent-control rows report maximum absolute mean changes across the reported checks; inconsistent-control rows report signed mean transfer changes for BACE/Llama/LoRA, with pointwise 95% intervals. The two groups summarize different sets of comparisons.

This supremum ranges over all linear scores, whereas measured D_k comes from a classifier fitted on finite data. The invariance statement concerns the supremum; whether a refitted LR classifier attains the same score is checked empirically. A low transferred score alone provides no low upper bound on this optimum.

Let $r \in \{S, T\}$ index shadow and target models and $c \in \{F, L\}$ index Frozen and LoRA. For a scoring-function family $\mathcal{F}$, define population restricted decodability as

$$\mathcal{D}^{\mathcal{F}}_{c,r} = \sup_{q \in \mathcal{F}} \mathrm{AUROC}_{(Z^r_c, a)} \left(q(Z^r_c), a\right). \tag{13}$$

For the target role and a linear family at fixed attribute k, this population quantity corresponds to $D^{*,\mathrm{lin}}_{T,k}$ in Equation 12, with condition c made explicit. The supremum over all measurable scores defines Bayes decodability; a linear family or a specified MLP architecture gives a restricted quantity. Our trained classifiers estimate algorithm-dependent performance rather than this supremum.

For a binary property with both classes having positive probability, write

$$\mathrm{AUROC}_P(q) = \mathbb{E}_{Z^+ \sim P(\cdot|a=1),\, Z^- \sim P(\cdot|a=0)} \left[\mathbf{1}\{q(Z^+) > q(Z^-)\} + \tfrac{1}{2}\mathbf{1}\{q(Z^+) = q(Z^-)\}\right].$$

The positive and negative examples in this definition are independent draws. This definition makes clear that AUROC depends on ranking and on class-conditional feature distributions, rather than on class prevalence alone.

Rotation invariance. For a row vector and an orthogonal matrix, the coordinate change has the explicit inverse

$$z' = zQ, \qquad Q^\top Q = I_d, \qquad z = z'Q^\top. \tag{14}$$

The equivalent-score construction in Equation 5 gives

$$D^{*,\mathrm{lin}}_{T,k}(Z_T Q) = D^{*,\mathrm{lin}}_{T,k}(Z_T). \tag{15}$$

For separate role rotations Q_S and Q_T, the equivalent shadow weight is $Q_S^\top w$, giving the transferred score

$$s'_{S \to T}(z_T) = z_T Q_T Q_S^\top w + b. \tag{16}$$

A common rotation makes $Q_T Q_S^\top = I_d$. Different rotations can change rankings but need not always reduce transfer. Whether refitting reaches the equivalent weights is checked by the common-rotation experiment in Section 4.4.

An invertible orthogonal transform preserves within-model optimal linear AUROC because every row-vector score $zw + b$ has an equivalent score $(zQ)(Q^\top w) + b$. Applying the transform only to target representations while keeping the shadow score fixed need not preserve transfer. This motivates the implemented controls in Section 3.4.

Proposition 1 (Consistent transforms and a fixed transferred score)**.** *Let Z be a row vector and Q an orthogonal matrix. The supremum of within-distribution AUROC over affine linear scores is identical for Z and ZQ. The supremum over all measurable scores is preserved by any measurable bijection with a measurable inverse. A fixed shadow-trained score need not retain its AUROC when only target representations are transformed.*

Proof. For $q(z) = zw + b$, define $q_Q(z') = z'Q^\top w + b$. Then $q_Q(zQ) = zQQ^\top w + b = q(z)$. The map on weights is bijective, so every ranking is available before and after transformation. For a general measurable bijection f, the map $q \mapsto q \circ f^{-1}$ likewise preserves all attainable rankings. For the final claim, take a balanced property a and scalar target $Z = 2a - 1$. The fixed score $q(z) = z$ has AUROC 1, but after target-only multiplication by -1 it has AUROC 0. Retraining the target-side linear score as $-z$ restores AUROC 1. This is an invertible transform with no attribute information loss. □

The first claim also holds for invertible linear maps with a compensating inverse weight map for row-vector scores. Orthogonal maps additionally preserve Euclidean norms, which makes them convenient controls with isotropic regularization. Finite optimization, standardization choices, LayerNorm, non-isotropic regularization, or restricted parameterization can break numerical invariance; the empirical controls test the implemented LR pipeline rather than assuming exact training invariance.

Proposition 2 (A sufficient condition for score transfer)**.** *Let P_S^+, P_S^- and P_T^+, P_T^- denote class-conditional representation distributions in a common coordinate system. For any fixed measurable score q,*

$$|\,\mathrm{AUROC}_T(q) - \mathrm{AUROC}_S(q)| \le \mathrm{TV}(P_T^+ \otimes P_T^-, P_S^+ \otimes P_S^-) \le \mathrm{TV}(P_T^+, P_S^+) + \mathrm{TV}(P_T^-, P_S^-),$$

where $\mathrm{TV}(P, Q) = \sup_B |P(B) - Q(B)|$.

Proof. The AUROC ranking kernel takes values in $[0, 1]$, so the difference of its expectations is bounded by total variation between the positive–negative pair distributions. Insert $P_T^+ \otimes P_S^-$, apply the triangle inequality, and use the fact that multiplying both measures by the same probability measure preserves their total-variation distance. □

If $\widehat{q}_S \in \mathcal{F}$ has population shadow excess AUROC at most ϵ, namely $\sup_{q\in\mathcal{F}} \mathrm{AUROC}_S(q) - \mathrm{AUROC}_S(\widehat{q}_S) \le \epsilon$, then

$$\sup_{q\in\mathcal{F}} \mathrm{AUROC}_T(q) - \mathrm{AUROC}_T(\widehat{q}_S) \le \epsilon + 2\delta, \quad \delta = \mathrm{TV}(P_T^+ \otimes P_T^-, P_S^+ \otimes P_S^-).$$

This follows by inserting the two shadow AUROCs around the target optimum and applying the preceding bound twice. It states a sufficient population condition, not an estimated finite-sample guarantee. Similar architecture alone does not bound δ, and our CKA or role-classifier scores are not estimators of this class-conditional pair distance. A saturated role classifier can distinguish marginal distributions without quantifying the difficulty of the particular attribute ranking.

Information boundary. A deeper state can make an attribute easier for a restricted classifier to classify without creating additional attribute information (Alain & Bengio, 2017; Belinkov, 2022). For a fixed trained model and deterministic input processing, $a \to G \to z_p \to h_\ell$ is a Markov chain, giving $I(a; h_\ell) \le I(a; z_p)$ (Cover & Thomas, 2006). For a full decoder-state chain, data processing holds across successive full states. A saved final-token vector alone need not be sufficient to compute the next layer's final-token vector, which also attends to preceding tokens; we therefore do not assert a general Markov chain over isolated token vectors. This distinction leaves intact the bound from the complete fixed-prompt decoder input and the claim that better linear decodability does not establish increased total information.

D Matched Optimization: Absolute, Individual-Property, and Seed Results

These results expand the matched study in Section 4.5. The within-shadow score is denoted W throughout the main text; S denotes only the shadow role.

Decodability and transfer differences. Under matched training, deepest-layer target-side GELU-MLP AUROC ranges from 0.647 to 0.756 on NCI1 and from 0.854 to 0.934 on BACE. Direct transfer ranges from 0.462 to 0.515 and from 0.484 to 0.538, respectively (Figure 7). Target-side GELU-MLP differences are -0.091 on NCI1/S1, -0.069 on BACE/S1, -0.040 on NCI1/Llama, and $+0.004$ on

BACE/Llama. The corresponding transfer differences are -0.041, $+0.034$, -0.050, and -0.044 (Table 8). On BACE/S1, for example, target-side decodability decreases while transfer increases. Adaptation therefore need not move the two measurements in the same direction. The absolute and within-shadow scores below belong to this matched study and are reported separately from the layer-wise study.

Table 8: Matched optimization: deepest-layer operational LoRA–Frozen differences.

Data	Decoder	Classifier	ΔA	Pointwise 95% CI	Holm p
NCI1	S1	LR	-0.012	[-0.073, 0.046]	1.000
		MLP	-0.041	[-0.086, 0.008]	0.714
BACE	S1	LR	-0.022	[-0.119, 0.048]	1.000
		MLP	+0.034	[-0.034, 0.102]	1.000
NCI1	Llama	LR	-0.046	[-0.072, -0.020]	0.113
		MLP	-0.050	[-0.117, 0.017]	1.000
BACE	Llama	LR	+0.015	[-0.080, 0.109]	1.000
		MLP	-0.044	[-0.101, 0.012]	0.714

Note. Hierarchical intervals are pointwise; Holm p values adjust the eight paired model-seed t tests. None passes the family-wise 0.05 threshold.

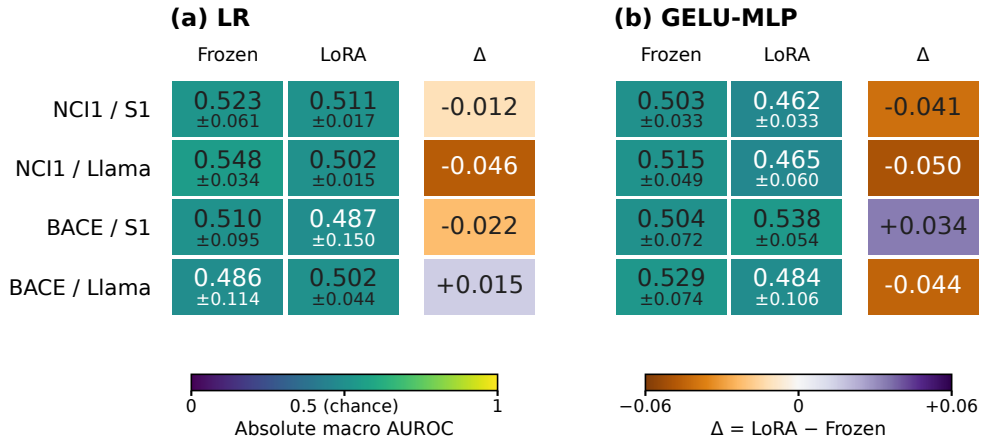


Figure 7: Matched optimization study: deepest-layer $S \to T$ macro AUROC. Frozen/LoRA cells show mean $\pm$ model-seed SD; Δ is LoRA minus Frozen. Each mean averages five structural properties and five task-model seeds; MLP first averages five classifier seeds within each model seed. Absolute scores and differences use separate color scales. Colors encode values, not significance. Table 8 retains the paired-difference intervals and Holm-adjusted tests; Figure 8 shows individual seed differences.

Sensitivity to adaptation configuration. In the focused NCI1/S1 sweep, rank changes do not produce a common trend across classifier families: at rank 16, transfer is slightly higher than Frozen for LR but lower for MLP. Restricting adapters to q/v projections leaves transfer lower for both classifiers. Restricting adaptation to the last quarter of decoder blocks gives opposite transfer directions for LR and MLP. Removing dropout and permuting task labels both retain lower transfer than Frozen for the two classifiers, while their magnitudes differ from the main configuration. Across these controls, target-side and transfer differences also vary by path and classifier (Appendix I.2). The sweep does not establish one adaptation direction or identify an optimal privacy configuration. CKA and update norms describe representation movement without isolating a causal privacy mechanism.

Statistical evidence across comparisons. None of the eight primary transfer comparisons survives Holm correction (Table 8), and none of the 840 exploratory comparisons survives BH-FDR at 0.05. All four MLP transfer intervals include zero. NCI1/Llama LR has a pointwise hierarchical interval below zero, yet its adjusted $p = 0.113$. The intervals describe uncertainty for individual comparisons, whereas the adjusted tests address the complete primary family. Together, these results leave heterogeneous, uncertain effect sizes: they establish neither a consistent LoRA transfer effect nor equivalence between Frozen and LoRA. Task noninferiority is established only for NCI1/Llama in the matched study and for no setting in the replication; detailed utility results appear in Appendices E.2 and F.4.

Table 9: Matched deepest-layer absolute macro AUROC, mean ± standard deviation across five model seeds.

Data	Decoder	Branch	D_{LR}	D_{MLP}	A_{LR}	A_{MLP}
NCI1	S1	F	0.706 ± 0.022	0.738 ± 0.025	0.523 ± 0.061	0.503 ± 0.033
		L	0.670 ± 0.048	0.647 ± 0.017	0.511 ± 0.017	0.462 ± 0.033
	Llama	F	0.721 ± 0.016	0.756 ± 0.008	0.548 ± 0.034	0.515 ± 0.049
		L	0.703 ± 0.027	0.715 ± 0.054	0.502 ± 0.015	0.465 ± 0.060
BACE	S1	F	0.916 ± 0.020	0.922 ± 0.012	0.510 ± 0.095	0.504 ± 0.072
		L	0.888 ± 0.037	0.854 ± 0.112	0.487 ± 0.150	0.538 ± 0.054
	Llama	F	0.927 ± 0.036	0.930 ± 0.036	0.486 ± 0.114	0.529 ± 0.074
		L	0.922 ± 0.035	0.934 ± 0.029	0.502 ± 0.044	0.484 ± 0.106

Note. MLP averages its five classifier seeds within each model seed. These are separate saved models from the layer-wise decomposition.

Table 10: Matched deepest-layer GELU-MLP AUROC for each structural property, averaged across model and classifier seeds.

Data	Decoder	Branch	Path	Nodes	Edges	Density	Diameter	Radius
NCI1	S1	F	$T \to T$	0.747	0.745	0.731	0.717	0.753
		F	$S \to T$	0.486	0.504	0.485	0.527	0.515
		L	$T \to T$	0.656	0.656	0.651	0.625	0.648
		L	$S \to T$	0.441	0.472	0.448	0.484	0.466
	Llama	F	$T \to T$	0.760	0.762	0.763	0.727	0.766
		F	$S \to T$	0.523	0.526	0.521	0.499	0.503
		L	$T \to T$	0.722	0.726	0.729	0.679	0.721
		L	$S \to T$	0.473	0.466	0.467	0.457	0.459
BACE	S1	F	$T \to T$	0.933	0.921	0.945	0.906	0.908
		F	$S \to T$	0.511	0.479	0.506	0.510	0.513
		L	$T \to T$	0.868	0.853	0.882	0.832	0.834
		L	$S \to T$	0.566	0.561	0.565	0.491	0.506
	Llama	F	$T \to T$	0.942	0.935	0.957	0.906	0.908
		F	$S \to T$	0.547	0.538	0.498	0.530	0.530
		L	$T \to T$	0.946	0.935	0.959	0.914	0.914
		L	$S \to T$	0.496	0.480	0.491	0.484	0.471

Note. Related properties are reported individually but are not treated as independent replications.

The absolute and seed tables help distinguish a uniform change from a mean dominated by a subset of fits. For example, BACE/S1 matched LoRA has appreciable across-seed target-side MLP variability, whereas BACE/Llama target-side scores remain high. This does not establish a general relationship between task performance and attribute decodability, because model identity, optimization, and data composition change jointly across settings.

Table 11: Matched primary operational LoRA–Frozen macro-AUROC differences for every model seed and exact two-sided sign-flip p values.

Data	Decoder	Classifier	42	43	44	45	46	p_{flip}
NCI1	S1	LR	-0.088	+0.021	+0.062	+0.026	-0.079	0.750
		MLP	+0.014	-0.041	-0.056	-0.020	-0.103	0.125
BACE	S1	LR	-0.192	+0.050	+0.013	+0.022	-0.005	0.938
		MLP	-0.015	-0.056	+0.084	+0.058	+0.099	0.312
NCI1	Llama	LR	-0.046	-0.030	-0.069	-0.015	-0.073	0.062
		MLP	-0.107	-0.048	+0.011	+0.043	-0.148	0.312
BACE	Llama	LR	-0.035	+0.111	-0.144	-0.007	+0.152	0.875
		MLP	+0.026	-0.063	-0.053	-0.031	-0.102	0.125

Note. The seed is the independent model-fitting replication unit.

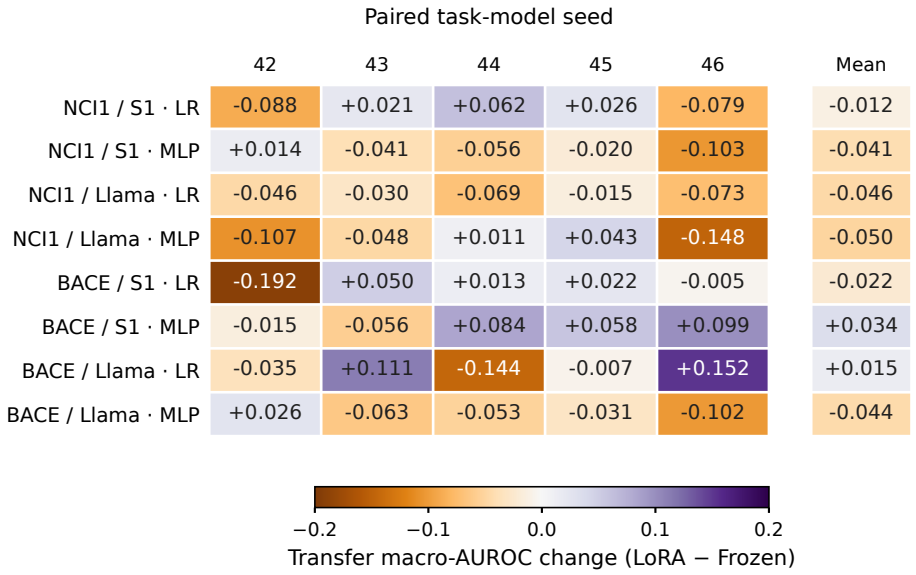


Figure 8: Seed-level transfer changes in the matched optimization study. Each cell is the paired LoRA–Frozen difference in deepest-layer $S \to T$ macro AUROC for one model seed, averaging the five properties and, for MLP, its five classifier seeds. The separate mean column averages the five model seeds. Signs and magnitudes remain visible when colors are unavailable. This descriptive matrix exposes heterogeneous seed outcomes; confirmatory inference uses the intervals and adjusted tests in Table 8.

E Uncertainty, Multiplicity, and Additional Attack Metrics

This appendix supplements the statistical conventions in Section 4.1 and the comparisons in Section 4.5. The primary Holm and exploratory BH families below test Frozen–LoRA differences. They are not inferential tests of the decodability–transfer gap $g = D - A$. Alignment and rotation intervals are separate pointwise analyses.

Table 12: Within-shadow reference for the matched optimization study: deepest-layer $S \to S$ macro AUROC, mean $\pm$ model-seed standard deviation.

Setting	Condition	LR	GELU-MLP
NCI1 / S1	Frozen	0.708 ± 0.038	0.728 ± 0.064
	LoRA	0.680 ± 0.062	0.691 ± 0.075
NCI1 / Llama	Frozen	0.737 ± 0.027	0.760 ± 0.008
	LoRA	0.721 ± 0.016	0.754 ± 0.018
BACE / S1	Frozen	0.924 ± 0.017	0.934 ± 0.017
	LoRA	0.865 ± 0.063	0.790 ± 0.180
BACE / Llama	Frozen	0.930 ± 0.019	0.939 ± 0.015
	LoRA	0.918 ± 0.032	0.920 ± 0.039

Note. MLP averages five classifier seeds within each model seed. These runs use the same fitting protocol as direct transfer, with the initialization distinction in Appendix A.3.

E.1 Units of analysis and confidence intervals

For scores $M_{1,i}$ and $M_{0,i}$ from two conditions at model seed i, define

$$d_i = M_{1,i} - M_{0,i}, \qquad \bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i. \tag{17}$$

For each model seed, we first compute AUROC separately for each property and classifier seed, average classifier seeds within that model seed, and then macro-average the five properties. LoRA–Frozen contrasts pair the same model seed, interface, property, graph IDs, and classifier seed. The five model seeds are the independent repetitions for model-seed inference; treating the 25 MLP fits as independent task-model repetitions would understate uncertainty.

The primary layer-wise and matched hierarchical intervals use 1,000 bootstrap replicates. The layer-wise and matched hierarchical bootstrap resamples paired model seeds, paired classifier seeds, and positive/negative target-test indices, retaining the same indices across the compared conditions. The retained implementation stratifies sample resampling separately within each property. This preserves Frozen/LoRA pairing but does not preserve all cross-property covariance induced by jointly sampling the same graph across the five labels. Its macro-property intervals are therefore a resampling approximation, not a full dependence-preserving graph bootstrap. We report them as pointwise diagnostics and base the family-wise decisions on paired model-seed tests of the already macro-averaged differences. This choice preserves property dependence within each observed model-seed endpoint. Uncertainty remains limited by the five seeds and overlapping benchmark graphs across repetitions.

Holm's procedure sorts the m paired-test p values in increasing order and compares them sequentially with

$$p_{(i)} \leq \frac{\alpha}{m - i + 1}, \qquad \alpha = 0.05. \tag{18}$$

Starting at the smallest p value, rejection stops at the first test that fails its threshold. The main-text primary families each contain eight comparisons; exploratory families use BH-FDR at 0.05. The pointwise alignment and rotation intervals are not part of these Holm families.

Within each of the matched and group-aware studies, the primary family contains eight deepest-layer transfer endpoints. Holm adjustment is applied to their paired t-test p values. The statistical protocol also reports exact sign-flip and sign tests. With only five nonzero paired differences, the smallest attainable two-sided sign-flip p value is $2/2^5 = 0.0625$; the exact test cannot attain 0.05 even with a perfectly consistent sign. Parametric t tests can yield smaller values but rely on stronger distributional assumptions. No primary endpoint passes Holm correction under the reported parametric analysis.

The matched study's exploratory BH family contains $4 \times 7 \times 3 \times 2 \times 5 = 840$ tests: four settings, seven interfaces, three paths, two classifiers, and five properties. All remain exploratory, with no

FDR-controlled rejection at 0.05. The configuration sweep is a distinct focused diagnostic family, rather than an opportunity to replace a nonsignificant primary configuration with a favorable post-hoc selection. Alignment and conditional-attack intervals likewise remain pointwise layer-wise diagnostics.

E.2 Task noninferiority and precision

For paired macro-F1 differences d_i and five seeds, the one-sided 95% lower bound is

$$L = \bar{d} - t_{0.95,4}\, s_d/\sqrt{5}.$$

Noninferiority at margin -0.02 is established if $L > -0.02$. These are per-setting claims, not a simultaneous declaration across four settings. A failure to establish noninferiority does not demonstrate a loss larger than 0.02. Mean closeness is likewise insufficient for equivalence, which would require an explicit two-sided equivalence analysis.

Table 13: Matched end-to-end task macro-F1 and per-setting noninferiority (NI) at margin -0.02.

Data	Decoder	Frozen	LoRA	Difference	Lower bound	NI
NCI1	S1	0.722	0.714	-0.007	-0.025	No
BACE	S1	0.706	0.704	-0.002	-0.025	No
NCI1	Llama	0.727	0.727	-0.000	-0.010	Yes
BACE	Llama	0.693	0.692	-0.001	-0.033	No

Note. The lower bound is one-sided 95%. A "No" entry means noninferiority was not established.

In the matched optimization study, mean target macro-F1 changes are -0.007, -0.002, approximately 0.000, and -0.001 for NCI1/S1, BACE/S1, NCI1/Llama, and BACE/Llama. Only NCI1/Llama establishes noninferiority (Figure 9). The group-aware replication establishes it in no setting (Appendix F.4).

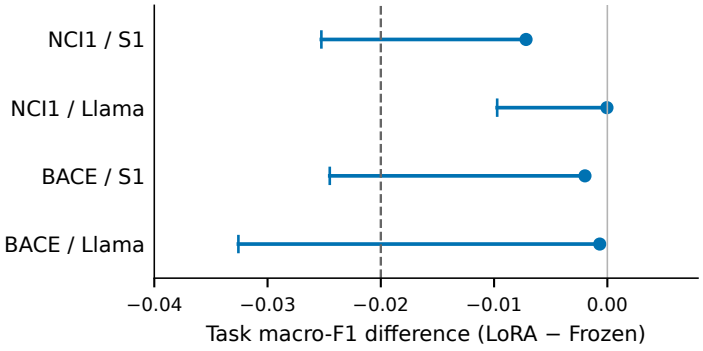


Figure 9: Task utility in the matched optimization study. Each dot is the mean LoRA–Frozen target task macro-F1 difference over five paired model seeds; the segment extends from the one-sided 95% lower confidence bound to that mean. The dashed line is the -0.02 noninferiority margin. Only NCI1/Llama has a lower bound above the margin.

The planning calculation for a meaningful absolute AUROC difference $\delta_0 = 0.03$ uses

$$n \approx \left\lceil \frac{(z_{1-\alpha/2} + z_{0.8})^2 s_d^2}{\delta_0^2} \right\rceil, \qquad \alpha = 0.05/8.$$

The result is sensitive to the variance estimated from five seeds. It provides a normal-approximation design diagnostic rather than an achieved-power guarantee or a recommendation to keep sampling until significance appears.

At 80% planned power and $\alpha = 0.05/8$, this calculation requires 9–201 independent model seeds to detect an absolute 0.03 AUROC difference, exceeding the five available seeds in every setting (Table 14). This assumes that the estimated paired-difference variance remains applicable to future independent model fits and uses a normal approximation despite estimating variance from only five seeds. It does not guarantee significance with additional data.

Table 14: Normal-approximation seed planning for the eight matched primary transfer endpoints.

Setting	**Classifier**	s_d	n
NCI1 / S1	LR	0.0676	65
	GELU-MLP	0.0436	28
NCI1 / Llama	LR	0.0250	9
	GELU-MLP	0.0796	90
BACE / S1	LR	0.0969	134
	GELU-MLP	0.0670	64
BACE / Llama	LR	0.1186	201
	GELU-MLP	0.0472	32

Note. s_d is the standard deviation of the five observed paired model-seed differences; n is the total independent model-seed count for the declared target effect and power.

E.3 Low-false-positive-rate and maximum-over-layer summaries

Table 15: Matched deepest-layer operational low-FPR sensitivity, replayed from saved classifier outputs.

Data	**Decoder**	**Branch**	**Classifier**	**TPR at FPR $\leq 1\%$**	**TPR at FPR $\leq 5\%$**
BACE	Llama	F	MLP	0.018	0.067
		F	LR	0.014	0.046
		L	MLP	0.023	0.071
		L	LR	0.012	0.028
	S1	F	MLP	0.014	0.051
		F	LR	0.018	0.054
		L	MLP	0.023	0.067
		L	LR	0.021	0.065
NCI1	Llama	F	MLP	0.008	0.036
		F	LR	0.007	0.045
		L	MLP	0.008	0.034
		L	LR	0.005	0.023
	S1	F	MLP	0.011	0.047
		F	LR	0.006	0.052
		L	MLP	0.009	0.038
		L	LR	0.007	0.031

Note. Each entry averages the empirical maximum TPR among ROC thresholds satisfying the stated FPR budget, first within model seed across properties and classifier seeds, then over five model seeds. This is a descriptive test-ROC statistic.

Both additions are descriptive audits of already observed results. For the low-FPR summary, the ROC curve is evaluated after the test set is opened; its threshold is not a deployable threshold selected on independent calibration data. At small test sizes, especially BACE, one false positive corresponds to a substantial FPR increment. The full AUROC and per-property sample counts remain necessary

Table 16: Layer-wise GELU-MLP maximum macro AUROC over the seven saved interfaces, computed separately per model seed and then summarized as mean ± standard deviation.

Data	Decoder	Branch	Max target decodability	Max transfer
BACE	Llama	F	0.937 ± 0.030	0.668 ± 0.069
		L	0.936 ± 0.030	0.677 ± 0.057
	S1	F	0.941 ± 0.024	0.651 ± 0.061
		L	0.936 ± 0.030	0.657 ± 0.061
NCI1	Llama	F	0.756 ± 0.014	0.552 ± 0.046
		L	0.753 ± 0.011	0.544 ± 0.044
	S1	F	0.765 ± 0.012	0.571 ± 0.037
		L	0.761 ± 0.008	0.565 ± 0.034

Note. This uses test-set layer selection and is descriptive only.

context. A maximum over test-set layer scores is similarly an observed upper envelope over seven interfaces, not an unbiased estimate of validation-selected deployment risk.

F Group-Aware Replication

This appendix documents the structure-based study in Section 4.5. Its estimates and test families remain separate from the layer-wise and matched studies. It repeats the three evaluation paths but not alignment or rotation.

F.1 Partitioning and the meaning of a clean holdout

The group-aware replication uses six partitions, targeting 32/8/10 percent of graphs for target train/validation/test and the same fractions for the shadow role. BACE uses Bemis–Murcko scaffold groups (Bemis & Murcko, 1996). NCI1 uses node-labeled one-iteration Weisfeiler–Lehman hash families (Shervashidze et al., 2011), which provide a conservative grouping heuristic rather than an exact molecular-identity certificate. All members of a group stay in one partition. The recorded allocation seed is $20260910 +$ model seed, distinct from the model-initialization seed. Across ten dataset/seed splits, the audit finds zero cross-partition group overlap and zero target–shadow sample overlap. BACE has 743 groups, with maximum size 55; NCI1 has 3,906 groups, with maximum size 4.

Table 17: Actual group-aware split sizes.

Data	Seed	T_{tr}	T_{val}	T_{test}	S_{tr}	S_{val}	S_{test}
BACE	42	484	121	151	485	121	151
	43	484	121	151	485	121	151
	44	485	121	151	484	121	151
	45	484	121	151	485	121	151
	46	485	121	151	484	121	151
NCI1	42	1316	328	411	1316	328	411
	43	1315	329	411	1315	329	411
	44	1315	328	411	1316	329	411
	45	1315	328	411	1316	329	411
	46	1315	329	411	1315	329	411

Note. Each row partitions the full filtered dataset; no sample is duplicated across the six partitions within a row.

Here, "clean" refers to final-test access discipline within the recorded replication: final-test loaders are not constructed until validation selects the relevant Phase-2 saved model. It does not mean the benchmark graphs were never encountered in any earlier experiment. A new allocation on an already studied benchmark is not an independently collected research population, and different seeds may share graphs across their separately allocated splits.

F.2 Corrected initialization and validation-only stability pilot

An earlier implementation omitted the shared classifier alignment between projector-space graph pretraining and post-decoder adaptation. Its model comparisons were excluded after training instability exposed the interface mismatch. In the retained protocol, graph-side Phase 1 is backbone-specific. The GNN/projector and projector-space head train for 100 epochs with validation selection; the selected encoder/projector is then passed through the frozen decoder, and a linear classifier trains for 10 epochs on cached post-decoder training states. Validation macro-F1 selects this shared post-decoder head. Frozen and LoRA both initialize from that same bundle within backbone, role, and model seed.

A separate validation-only pilot with model seed 41 was used to check numerical stability before opening the retained confirmatory final tests. The matched study's 5×10^{-4} learning rate diverged under this retrained interface, and 10^{-4} was unstable. The lowest attempted stable rate, 5×10^{-5}, was therefore locked for both branches. Seed 41 contributes to none of the reported estimates. This pilot is separate by model seed, not a claim of a wholly separate pool of benchmark graphs.

Phase 2 uses AdamW at 5×10^{-5}, weight decay 10^{-5}, batch size 32, clipping 0.5, and 30 epochs. The retained LoRA configuration is rank 8, $\alpha = 32$, dropout 0.1, on $q/k/v/o$ in every block. Each final-test membership is accessed in one combined evaluation/extraction pass after selecting the saved model; attacks are subsequently fitted and scored on the saved representations. The grouping, retrained graph side, aligned classifier initialization, and learning rate all differ from the matched optimization study, so their effects cannot be separated by comparing the aggregate means alone.

F.3 Complete validation and primary results

The completed gate admits 40 Phase-1 runs with zero final-test access, 80 Phase-2 runs with one post-selection final-test access each, 720 attack manifests, and 3,600 property-level rows. There are no classifier output-hash, sample-ID pairing, or metric-replay failures. Sixteen paired bootstrap endpoints each retain 1,000 valid replicates. The bootstrap resamples paired model seeds, paired classifier initializations, and label-stratified target-test sample IDs. Sample resampling is performed separately within each property, so the cross-property dependence limitation described in Appendix E also applies to this study. All 40 LoRA adapter configurations match the declared rank, scaling, and dropout. The primary family has the same eight deepest-layer operational endpoints as the matched analysis; the exploratory family has 120 deepest-layer property/path/classifier endpoints. The matched-study all-layer family has 840 endpoints because it includes seven interfaces.

Table 18: Group-aware replication primary operational LoRA–Frozen macro-AUROC differences.

Data	Decoder	Classifier	ΔA	Pointwise 95% CI	Holm p
NCI1	S1	LR	-0.010	[-0.044, 0.021]	1.000
		MLP	+0.003	[-0.027, 0.029]	1.000
BACE	S1	LR	+0.004	[-0.040, 0.039]	1.000
		MLP	-0.001	[-0.058, 0.060]	1.000
NCI1	Llama	LR	-0.011	[-0.041, 0.016]	1.000
		MLP	-0.016	[-0.040, 0.009]	0.623
BACE	Llama	LR	-0.053	[-0.100, -0.002]	0.815
		MLP	-0.131	[-0.295, 0.002]	1.000

Note. Intervals are pointwise hierarchical 95% intervals; Holm adjusts eight paired model-seed t tests.

No primary endpoint passes Holm correction, and no exploratory endpoint passes BH adjustment. BACE/Llama LR has a negative pointwise interval, but its adjusted $p = 0.815$. As in the matched study, this is not a family-wise-confirmed decrease. The BACE/Llama MLP mean is -0.131, with interval $[-0.295, 0.002]$; one seed contributes a difference of -0.437. Full seed reporting is particularly important for that endpoint.

Table 19: Group-aware replication primary differences for each model seed.

Data	Decoder	Classifier	42	43	44	45	46
NCI1	S1	LR	-0.058	-0.007	+0.039	-0.016	-0.007
		MLP	+0.008	+0.014	-0.043	+0.008	+0.028
BACE	S1	LR	+0.016	-0.043	+0.055	-0.029	+0.020
		MLP	-0.051	+0.004	-0.028	+0.095	-0.025
NCI1	Llama	LR	-0.027	+0.005	-0.014	+0.020	-0.041
		MLP	-0.019	+0.010	-0.022	-0.028	-0.022
BACE	Llama	LR	-0.108	-0.002	-0.065	-0.110	+0.019
		MLP	-0.203	+0.034	-0.056	-0.437	+0.006

F.4 Absolute decodability, transfer, and utility

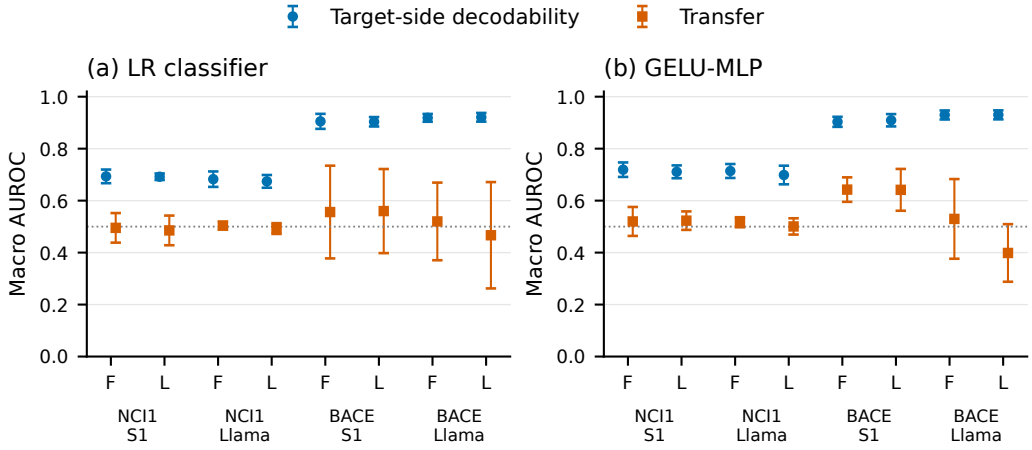


Figure 10: Group-aware replication deepest-layer decodability and transfer. F/L denote Frozen/LoRA. Error bars are standard deviations across five model seeds after macro averaging properties and, for MLP, classifier seeds. The full AUROC range is shown.

BACE target-side MLP AUROC remains approximately 0.90–0.93, while its operational transfer scores vary more widely. NCI1 target-side MLP scores are approximately 0.70–0.72 and transfer approximately 0.50–0.52. These absolute measurements support the distinction between decodability and transfer across the group-aware protocol. They do not establish invariant effect sizes across splits. In particular, AUROC below 0.5 is the performance of the fixed transferred score orientation; flipping its sign after inspecting target-test labels would grant information not available to that attack. A calibrated classifier could have a different orientation and performance.

Mean task macro-F1 declines by 0.010–0.018 across the four settings. Every one-sided lower bound falls below -0.02. This is insufficient evidence for noninferiority, not proof that the true loss exceeds the margin in every setting. The negative or inconclusive LoRA differences should therefore not be presented as a privacy–utility improvement.

Table 20: Group-aware replication absolute macro AUROC, mean ± model-seed standard deviation.

Data	Decoder	Branch	D_{LR}	D_{MLP}	A_{LR}	A_{MLP}
NCI1	S1	F	0.693 ± 0.026	0.719 ± 0.028	0.495 ± 0.057	0.520 ± 0.056
		L	0.692 ± 0.013	0.711 ± 0.025	0.485 ± 0.057	0.523 ± 0.036
	Llama	F	0.683 ± 0.030	0.714 ± 0.027	0.504 ± 0.015	0.517 ± 0.019
		L	0.674 ± 0.024	0.699 ± 0.036	0.493 ± 0.021	0.501 ± 0.031
BACE	S1	F	0.905 ± 0.029	0.903 ± 0.019	0.556 ± 0.178	0.642 ± 0.047
		L	0.903 ± 0.018	0.909 ± 0.024	0.560 ± 0.162	0.641 ± 0.080
	Llama	F	0.919 ± 0.015	0.930 ± 0.017	0.520 ± 0.149	0.530 ± 0.153
		L	0.921 ± 0.017	0.930 ± 0.017	0.467 ± 0.205	0.399 ± 0.111

Table 21: Group-aware replication end-to-end macro-F1 and per-setting noninferiority (NI) at -0.02.

Data	Decoder	Frozen	LoRA	Difference	Lower bound	NI
NCI1	S1	0.751	0.738	-0.012	-0.027	No
BACE	S1	0.746	0.728	-0.018	-0.081	No
NCI1	Llama	0.742	0.730	-0.012	-0.041	No
BACE	Llama	0.727	0.717	-0.010	-0.064	No

Note. Lower bounds are one-sided 95%. No setting establishes noninferiority.

G Reproducibility, Artifact Scope, and Code Provenance

G.1 Evidence records and identity checks

The layer-wise confound gate reports 1,200 baseline rows, 4,800 task-conditioned rows, 2,400 residual rows, 720 chemical rows, 1,680 classifier output files, and 320 bootstrap endpoints. The alignment gate reports 280 alignment cells, 8,400 diagnostic attack rows, 1,680 classifier output/saved model manifests, and 96 bootstrap endpoints. Their recorded validation reports have zero identity, hash, and AUROC-replay failures.

The matched study records 140 completed training runs, 8,820 classifier-output manifests, and 44,100 property-level rows. Its validation report records no failures in 600 shared-anchor comparisons or output-hash checks, and a maximum AUROC replay error of zero.

These validation counts describe checks against the saved experimental records. Reproducing the experiments requires the training and evaluation code, the public datasets, the specified base models, and the study-specific protocols documented in Appendices A–F. Code availability and access to the original saved representations, model checkpoints, and per-example predictions are distinct requirements: the latter are needed to replay the original runs exactly, whereas rerunning the implementation produces new experimental runs.

G.2 Environment and runtime accounting

The recorded experiment environment is Python 3.13.12, PyTorch 2.11.0 with CUDA 13.0, PyG 2.7.0, Transformers 5.9.0, PEFT 0.19.1, scikit-learn 1.8.0, NumPy 2.4.4, and pandas 3.0.2. The recorded GPUs are NVIDIA RTX 4090 devices with approximately 24 GB memory; the environment audit records driver 580.126.20. These versions describe the recorded experiment environment.

Across the 140 matched training manifests, recorded per-run elapsed durations sum to 9.074 hours. This is a sum of instrumented run durations, not a reconstruction of all project GPU-hours: it excludes portions outside each timer, layer-wise development, upstream graph-side training where separately executed, classifier-based evaluation, failures, and unrecorded runs. Parallel execution also means it

is not the calendar duration. The available records do not support a total GPU-hour estimate for the layer-wise study.

The layer-wise base saved models were named by repository identifier without an immutable upstream revision. S1's model card specifies Apache-2.0; Llama uses the Llama 3.1 community license. The actual model configuration, tokenizer convention, and locally recorded trained-saved model hashes provide additional identity evidence, but repository names do not retrospectively prove an upstream hub commit. The recorded Git HEAD also coexists with working-tree changes; the recorded file-level hashes identify the audited implementation more precisely than that commit alone.

G.3 Code lineage and release boundary

Our implementation is based on the publicly available code of Zhang et al. (2022). It retains or adapts their TU dataset readers, conventional GNN components, graph-statistic utilities, and original attribute-classifier components. We extend this implementation with a continuous graph-token interface to language models, layer-wise extraction, Frozen/LoRA comparisons, target-side and cross-model evaluation paths, orthogonal-alignment and rotation controls, matched-training and group-aware studies, and result-validation scripts. The implementation is therefore an extension of that codebase, rather than an implementation written entirely from scratch.

The upstream code and our extensions are distinguished by their roles in the experiment pipeline. The supplementary code is intended to support the training, representation extraction, and audit procedures described in this paper; code provenance remains relevant to both retained and modified components. The public datasets and pretrained language models are obtained from their original providers.

G.4 Boundaries of reproducibility and external validity

The reported results are indexed by study, interface, evaluation path, model seed, and property. Remaining limits concern upstream weight revision, full layer-wise compute accounting, and the independent reproduction of the original training runs. Layer-wise OVCAR-8H and AIDS stress tests are excluded from the primary evidence: the former used a with-replacement view with severe task imbalance, and the latter approached structural saturation. They do not expand the effective independent sample size of the two-dataset study. A published multi-token or generative graph-language pipeline, stronger property classifier, or alternative split policy defines a new evaluation domain rather than a result inferred from this audit.

H Connections to Transfer, Classifier-based Evaluation, and Benchmark Design

Within-shadow prediction and target transfer. Domain-adaptation theory separates source classification error from the discrepancies and joint classification assumptions needed to control target error (Ben-David et al., 2010). This motivates reporting the within-shadow reference together with transfer: source-side predictive success alone does not determine target performance. The cited bounds concern classification error under their assumptions; our AUROC measurements and the pair-distribution statement in Appendix C.3 use a different evaluation quantity. We do not estimate a domain-adaptation bound from the observed role-classifier accuracy.

Alignment and available target data. Subspace alignment learns a relation between source and target feature spaces (Fernando et al., 2013); domain-adversarial training learns features that support a task while reducing domain distinguishability (Ganin et al., 2016). These approaches illustrate different ways in which representation compatibility enters transfer. Our diagnostic fixes the task models and estimates an orthogonal map from paired validation representations. It requires graph correspondences that general unpaired domain-adaptation methods need not receive. Conversely, we do not retrain task representations with a domain-adversarial objective. Common preprocessing is fitted on the same paired validation budget in both aligned and unaligned comparisons, isolating the additional orthogonal mapping step.

Similarity and attribute-specific classifier output. SVCCA and related canonical-correlation analyses compare representations across networks and training stages (Raghu et al., 2017; Morcos et al., 2018). CKA provides a complementary representation-similarity measure (Kornblith et al., 2019). These quantities summarize feature relationships under different invariances. A transferred attribute score additionally depends on the classifier direction and the attribute-conditioned representation distributions. Our CKA and model-origin results in Appendix I.4 are therefore interpreted alongside measured transfer, rather than as interchangeable exposure scores. SVCCA and canonical-correlation comparisons are background methods, not additional experiments in this audit.

What a classifier result establishes. Information-theoretic classifier-based evaluation and minimum-description-length classifier-based evaluation evaluate representations through objectives that differ from raw held-out classification scores (Pimentel et al., 2020; Voita & Titov, 2020). Conditional classifier-based evaluation explicitly asks what remains accessible beyond a baseline (Hewitt et al., 2021). Our audit instead reports AUROC from the declared LR and MLP fitting procedures, together with task-conditioned and residual-label controls. Those measurements do not estimate mutual information, description length, or optimal performance over all classifiers. INLP and LEACE study interventions aimed at linear attribute removal (Ravfogel et al., 2020; Belrose et al., 2023); they help distinguish an explicit removal objective from the transfer changes observed after task adaptation.

Benchmark and population scope. Open Graph Benchmark and the benchmarking framework of Dwivedi et al. (2023) emphasize evaluation protocols that support reproducible graph learning comparisons (Hu et al., 2020). The present study uses the declared NCI1 and BACE populations rather than adding their benchmark datasets. Scaffold and WL grouping specify which structural families stay together; they do not certify separation of every chemically or statistically similar pair. Repartitioning a previously studied benchmark improves within-study separation without creating a new research population. These considerations explain why the three experimental groups are reported separately.

Observation channels in property inference. Collaborative-learning attacks recover unintended properties from shared training updates (Melis et al., 2019). Our observer instead receives a per-input graph representation and predicts an attribute of that input. The exposure channel, unit of inference, and available labels differ. Stating those differences prevents a result about one interface from being read as a claim about all forms of model or training-data privacy.

I Supporting Results for the Three Studies

I.1 Layer-wise within-shadow reference

The notation W refers to the within-shadow score, while S identifies the shadow model. Table 22 reports the layer-wise deepest-layer $S \rightarrow S$ reference. Means and standard deviations are across five task-model seeds; property and classifier-seed scores are first averaged within each model seed. These saved models differ from the matched-study reference in Appendix D. Together with Table 1, these scores show strong within-model attribute prediction in both roles despite weaker direct transfer.

I.2 Matched configuration contrasts

Table 23 gives deepest-layer NCI1/S1 contrasts against Frozen for the supplementary configurations. They are descriptive comparisons of existing runs, not additional experiments or replacement primary tests. Rank-4 and rank-16 controls vary rank; the remaining rows retain rank 8. The exact adapter matrix is in Appendix A.2. These exploratory contrasts retain their original comparator and are separate from the primary test family.

Table 22: Layer-wise deepest-layer within-shadow macro AUROC, mean ± model-seed SD.

Data	Decoder	Branch	LR	MLP
NCI1	S1	F	0.708 ± 0.038	0.728 ± 0.068
		L	0.708 ± 0.028	0.706 ± 0.051
	Llama	F	0.737 ± 0.027	0.765 ± 0.011
		L	0.711 ± 0.019	0.728 ± 0.052
BACE	S1	F	0.924 ± 0.017	0.933 ± 0.016
		L	0.904 ± 0.032	0.924 ± 0.024
	Llama	F	0.930 ± 0.019	0.938 ± 0.016
		L	0.925 ± 0.026	0.932 ± 0.029

Table 23: NCI1/S1 supplementary configuration minus Frozen: deepest-layer target-side and transfer AUROC differences.

Configuration	ΔD_{LR}	ΔD_{MLP}	ΔA_{LR}	ΔA_{MLP}
Rank 4	-0.016	-0.046	-0.013	-0.028
Rank 16	-0.057	-0.104	+0.001	-0.008
q/v only	-0.003	-0.032	-0.024	-0.004
Last quarter	-0.006	-0.003	-0.005	+0.014
Dropout 0	-0.043	-0.098	-0.019	-0.016
Permuted task labels	-0.066	-0.088	-0.025	-0.030

I.3 Replication within-shadow reference

Table 24 complements the target-side and transfer scores in Appendix F.4. In BACE, the shadow-side MLP score stays close to 0.89 for both decoders and conditions, while transfer varies substantially. This reference describes the new group-aware protocol only.

Table 24: Structure-based replication deepest-layer within-shadow macro AUROC, mean ± model-seed SD.

Data	Decoder	Branch	LR	MLP
NCI1	S1	F	0.691 ± 0.020	0.723 ± 0.032
		L	0.691 ± 0.028	0.721 ± 0.032
BACE	S1	F	0.884 ± 0.035	0.887 ± 0.036
		L	0.888 ± 0.029	0.888 ± 0.036
NCI1	Llama	F	0.718 ± 0.009	0.742 ± 0.006
		L	0.705 ± 0.011	0.730 ± 0.006
BACE	Llama	F	0.892 ± 0.035	0.892 ± 0.026
		L	0.885 ± 0.039	0.892 ± 0.027

I.4 Layer-wise representation geometry

Across pooled settings, correlations between CKA and the layer-wise $A - D$ difference are -0.335 for LR and -0.320 for MLP, but their signs vary within settings. A model-origin classifier frequently distinguishes shadow from target representations nearly perfectly. Table 25 summarizes the full range over the seven interfaces and five seeds in each dataset/decoder/branch group. The role-

classification diagnostic is frequently saturated even when CKA varies. Table 26 reports the recorded CKA associations with the layer-wise $A - D$ difference, including the separate setting estimates. The change of sign across settings limits interpretation of the pooled negative association. Neither diagnostic measures the class-conditional ranking compatibility required by a particular attribute classifier.

Table 25: Layer-wise geometry diagnostics: ranges across all seven interfaces and five model seeds per setting and branch.

Data	Decoder	Branch	CKA range	Role AUROC range
BACE	Llama	F	0.187–0.608	1.000–1.000
		L	0.187–0.645	1.000–1.000
	S1	F	0.007–0.606	1.000–1.000
		L	0.122–0.606	1.000–1.000
NCI1	Llama	F	0.147–0.459	1.000–1.000
		L	0.177–0.462	1.000–1.000
	S1	F	0.036–0.602	0.994–1.000
		L	0.005–0.602	0.996–1.000

Table 26: CKA association with the layer-wise transfer-minus-decodability difference.

Scope	Classifier	Spearman ρ	Pointwise 95% CI
overall	LR	-0.335	[-0.504, -0.161]
nci1_s1	LR	-0.012	[-0.328, +0.339]
bace_s1	LR	+0.058	[-0.230, +0.282]
nci1_llama	LR	-0.270	[-0.543, +0.152]
bace_llama	LR	-0.424	[-0.661, -0.046]
overall	MLP	-0.320	[-0.475, -0.126]
nci1_s1	MLP	-0.251	[-0.448, -0.066]
bace_s1	MLP	-0.038	[-0.350, +0.158]
nci1_llama	MLP	+0.072	[-0.502, +0.497]
bace_llama	MLP	-0.423	[-0.605, +0.030]

Note. Intervals are the recorded pointwise bootstrap intervals, without multiplicity adjustment.

The tables in this appendix keep the layer-wise, matched-training, and group-aware results separate. Each summary uses the records from its corresponding study.

I.5 Matched and structure-based validation summaries

Tables 27 and 28 collect the final-layer MLP results used in Section 4.5. Each study uses its own training and partitioning protocol. The three paths compare target-side decodability, direct transfer, and within-shadow classification; the LoRA–Frozen contrasts assess adaptation effects within each study. Matched-training details are in Appendix A.2, and structure-based training and split details are in Appendix F.

In the structure-based study, BACE/Llama within-shadow MLP macro AUROC is 0.892 under both Frozen and LoRA; BACE/S1 scores are 0.887 and 0.888. The four MLP LoRA–Frozen transfer

Table 27: Deepest-layer MLP macro AUROC under matched training and structure-based grouping, mean ± model-seed SD.

Study	Data	Decoder	Condition	D	A	W
Matched	NCI1	S1	F	0.738 ± 0.025	0.503 ± 0.033	0.728 ± 0.064
		S1	L	0.647 ± 0.017	0.462 ± 0.033	0.691 ± 0.075
		Llama	F	0.756 ± 0.008	0.515 ± 0.049	0.760 ± 0.008
		Llama	L	0.715 ± 0.054	0.465 ± 0.060	0.754 ± 0.018
	BACE	S1	F	0.922 ± 0.012	0.504 ± 0.072	0.934 ± 0.017
		S1	L	0.854 ± 0.112	0.538 ± 0.054	0.790 ± 0.180
		Llama	F	0.930 ± 0.036	0.529 ± 0.074	0.939 ± 0.015
		Llama	L	0.934 ± 0.029	0.484 ± 0.106	0.920 ± 0.039
Grouped	NCI1	S1	F	0.719 ± 0.028	0.520 ± 0.056	0.723 ± 0.032
		S1	L	0.711 ± 0.025	0.523 ± 0.036	0.721 ± 0.032
		Llama	F	0.714 ± 0.027	0.517 ± 0.019	0.742 ± 0.006
		Llama	L	0.699 ± 0.036	0.501 ± 0.031	0.730 ± 0.006
	BACE	S1	F	0.903 ± 0.019	0.642 ± 0.047	0.887 ± 0.036
		S1	L	0.909 ± 0.024	0.641 ± 0.080	0.888 ± 0.036
		Llama	F	0.930 ± 0.017	0.530 ± 0.153	0.892 ± 0.026
		Llama	L	0.930 ± 0.017	0.399 ± 0.111	0.892 ± 0.027

Note. D, A, and W denote $T \to T$, $S \to T$, and $S \to S$; F/L denote Frozen/LoRA. Each study is summarized separately, using five model seeds after averaging classifier seeds and properties. Full LR and MLP results appear in the appendices.

Table 28: Deepest-layer MLP direct-transfer differences, LoRA minus Frozen.

Study	Data	Decoder	ΔA	Pointwise 95% CI	Holm p
Matched	NCI1	S1	-0.041	[-0.086, 0.008]	0.714
	BACE	S1	+0.034	[-0.034, 0.102]	1.000
	NCI1	Llama	-0.050	[-0.117, 0.017]	1.000
	BACE	Llama	-0.044	[-0.101, 0.012]	0.714
Grouped	NCI1	S1	+0.003	[-0.027, 0.029]	1.000
	BACE	S1	-0.001	[-0.058, 0.060]	1.000
	NCI1	Llama	-0.016	[-0.040, 0.009]	0.623
	BACE	Llama	-0.131	[-0.295, 0.002]	1.000

Note. Intervals are pointwise paired hierarchical 95% intervals. Holm p values are adjusted within each study's full family of eight LR/MLP comparisons, not only the four MLP rows shown. Complete primary tests and seed results are retained in the appendices.

differences are $+0.003$, -0.001, -0.016, and -0.131 for NCI1/S1, BACE/S1, NCI1/Llama, and BACE/Llama, respectively. All four pointwise intervals include zero. These summaries retain the within-shadow reference and setting-specific contrasts alongside the shorter account in Section 4.5.