HARD LATENT-TABLE ADAPTATION FROM LINK PREDICTION TO SUBGRAPH TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Link-prediction pretraining supplies node representations for subgraph classification. Adapting these representations, however, requires subgraph-level labels to update a node table shared by overlapping examples. These labels constrain aggregated predictions, leaving the individual node updates only indirectly supervised. We propose HardLT, a latent-table adaptation objective that couples complete-view task fitting with supervised training on globally coherent partial views. It constructs these views by masking fixed PCA directions jointly across all nodes before aggregation, then selects one common mask using the minibatch's mean classification loss. The complete view fits the task, while the selected view supplies an additional constraint on adaptation. Dataset-specific configurations also align the two predictions through a consistency loss. With the GPEN encoder, HardLT achieves the highest mean scores on HPO-Metab, EM-User, and HPO-Neuro among the compared configurations, including a tie with GPEN on HPO-Neuro. In matched component studies, HardLT raises mean micro-F1 from 0.634 to 0.647 on HPO-Metab and from 0.916 to 0.922 on EM-User. Global sharing gives higher means than nodewise perturbations in both studies. Fixed-view evaluation on the same HPO-Metab checkpoints shows smaller performance decreases under shared latent-direction perturbations. HardLT adds no parameters or computation at inference.

1 INTRODUCTION

Subgraph classification predicts properties of functional modules in protein networks, disease-associated gene sets, and communities in social networks. Subgraphs with similar local structures can occupy different positions in the surrounding graph. Existing methods capture this context through structural encoders, node labeling, neighborhood sampling, and auxiliary pretraining (Alsentzer et al., 2020; Wang & Zhang, 2022; Kim & Oh, 2024; Jacob et al., 2023; Kim & Oh, 2025; Zhong et al., 2025). GPEN (Wu et al., 2025), for example, uses pretrained node features for subgraph classification. Prior graph-pretraining work also shows that the level of pretraining supervision affects downstream transfer (Hu et al., 2020). We focus on adapting node tables pretrained by link prediction (LP).

Pretraining and downstream adaptation supervise this table at different granularities. LP observes relations between node pairs; a subgraph label supervises a prediction aggregated from many node vectors. The label does not specify which vectors should change, and changing one vector affects every labeled subgraph that contains that node. For example, two overlapping subgraphs can send different task signals to the same table row. We hypothesize that complete-view fitting alone leaves adaptation underdetermined: models with similar training loss may rely on different, potentially fragile combinations of latent directions. HardLT addresses this sensitivity by training the shared table under task-selected transformations as well as its complete view.

This setting raises two design questions. First, how should a view be defined when examples query shared node parameters? Applying a common transformation before aggregation lets all subgraphs in a minibatch query one perturbed table. Second, which views should guide adaptation? A random view may leave the current predictions largely unchanged, whereas the downstream classification loss can identify sampled transformations under which fitting is more difficult. We retain the complete table in training to fit the representation used at inference directly.

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.

We propose HardLT to train the classifier on both the complete table and a difficult partial view. A PCA basis fitted once to the initial table defines the reference coordinates. At each minibatch, HardLT samples K masks, each retaining a fixed number of coordinates, and applies each mask to every node. It selects one view by the minibatch's mean classification loss and trains on that view together with the complete table. The complete-view term fits the task, while the partial-view term penalizes errors exposed by the selected transformation. Dataset-specific configurations also use prediction consistency between the two branches. The table and classifier are updated jointly; inference uses only the complete table.

HardLT formulates view construction and selection at the level of the shared parameter table. Building on feature-direction perturbations (Rahmani & Atia, 2017) and loss-based view selection (Gong et al., 2021; Ferreira et al., 2025), it applies one common transformation before subgraph aggregation and selects it jointly for the minibatch. Global sharing couples the transformations of distinct node rows, beyond reusing a nodewise mask across occurrences of the same node. This design connects the scope of the perturbation to the scope of the adapted parameters, while the downstream labels determine which sampled transformation supplies the auxiliary update.

We evaluate these choices through benchmark, component, and sensitivity studies. Among the configurations in Table 1, HardLT achieves the highest mean scores on HPO-Metab, EM-User, and HPO-Neuro, including a tie with GPEN on HPO-Neuro, using the GPEN encoder throughout. Matched component studies compare changes to its adaptation objective: HardLT raises the mean by 0.013 on HPO-Metab and 0.006 on EM-User over complete-view adaptation. Shared transformations have higher means than nodewise transformations in both studies; the comparison with view averaging varies by task. A sensitivity study varies candidate count, retention budget, and auxiliary weight. Evaluating the same HPO-Metab checkpoints under fixed shared views further shows smaller performance decreases for HardLT across all five tested retention budgets (Figure 3). The operator analysis characterizes cross-row covariance and the selection constraint, connecting these design choices to the shared-table setting.

Our contributions are:

- We formulate view regularization on a trainable pretrained node table reused by overlapping subgraphs, motivated by the change from edge-level to subgraph-level supervision.
- We develop HardLT, combining globally shared direction masks, minibatch-level loss selection, and complete-view training. Our analysis characterizes the cross-row covariance induced by sharing and the loss-tail weighting induced by candidate selection.
- HardLT achieves the highest mean scores on three of four benchmarks among the configurations in Table 1, including a tie with GPEN on HPO-Neuro. Matched component studies on HPO-Metab and EM-User support shared-view adaptation; fixed-view evaluation on HPO-Metab further shows reduced perturbation sensitivity. Inference requires no additional computation.

2 Related work

Subgraph representation learning. Sub2Vec learns unsupervised representations of subgraphs (Adhikari et al., 2018). SubGNN aggregates subgraph components and anchor patches (Alsentzer et al., 2020), while GLASS distinguishes target nodes through labeling (Wang & Zhang, 2022). S2N represents subgraphs as nodes in a derived graph (Kim & Oh, 2024); WLKS and ISNN develop further structural representations (Kim & Oh, 2025; Zhong et al., 2025). GPEN combines global position encoding with subgraph encoding and uses pretrained node representations (Wu et al., 2025). Other methods construct multiple views through subgraph augmentation (Shen et al., 2022) or random-walk neighborhood sampling (Jacob et al., 2023). HardLT acts on the pretrained node table before aggregation, retaining the host's graph structure and subgraph membership. Our experiments study this adaptation objective with the GPEN encoder; WLKS and ISNN provide complementary benchmark comparisons through their own representation constructions.

Graph pretraining and augmentation. Node2vec learns node representations from sampled network neighborhoods (Grover & Leskovec, 2016), while graph-pretraining strategies combine node- and graph-level objectives (Hu et al., 2020). GraphCL constructs augmented graph pairs for contrastive learning (You et al., 2020); DropEdge regularizes graph convolution by randomly

deleting edges during training (Rong et al., 2020). HardLT operates during supervised adaptation: it transforms the current node table in a fixed reference basis, with graph connectivity and subgraph membership held fixed.

Dropout and difficult views. Dropout regularizes training by randomly dropping neural units (Srivastava et al., 2014). Generalized Dropout introduces learnable dropout parameters (Srinivas & Babu, 2016), and basis dropout removes directions in an arbitrary orthogonal basis (Rahmani & Atia, 2017); its CNN construction can share a channel projection across spatial positions. MaxUp selects a largest-loss augmentation per example (Gong et al., 2021). Hard View Pretraining (HVP) scores augmentation pairs per image before training on the selected pairs (Ferreira et al., 2025). HardLT builds on these established operations to regularize an adapting parameter table reused across subgraphs, with a single selected transformation for the minibatch. The comparison below describes the cited formulations, rather than all possible extensions; "clean loss" denotes a separate unperturbed-view objective.

Design axis	**Basis dropout**	**MaxUp**	**HVP**	**HardLT**
Perturbed object	Inputs or activations	Training examples	Image views	Shared node parameters
Reference basis	Orthogonal	Not required	Not required	Fixed PCA of E_0
Sharing scope	CNN spatial positions	Per example	Per image	All rows of the table
View selection	Random	Per-example max	Max-loss pair	Batch-mean max
Clean loss	No	No	No	Yes
Trained variables	Network	Network	SSL network	Table and predictor

Perturbing pretrained representations. HiddenCut removes spans of hidden representations during language-model fine-tuning (Chen et al., 2021), and NEFTune perturbs embeddings during instruction fine-tuning (Jain et al., 2024). Coordinate-removal studies also examine how text embeddings retain information under feature removal (Takeshita et al., 2025). In subgraph classification, a table row can participate in several overlapping examples. Sharing a mask across rows therefore couples the perturbations seen by different nodes and subgraphs; independent nodewise masks do not introduce the same coupling.

Prediction consistency. R-Drop aligns predictions from two dropout passes using bidirectional KL divergence (Liang et al., 2021). HardLT's optional consistency term uses a detached complete-view prediction as the target for a partial view. Its candidate selection depends on supervised classification loss; consistency is applied during the selected view's training replay.

Regularizing pretrained parameters. L2-SP penalizes distance from pretrained parameters (Li et al., 2018). HardLT instead penalizes classification loss under sampled transformations of the current table. Its auxiliary term measures task loss on transformed representations rather than parameter distance to initialization.

3 Problem formulation

Let $G = (\mathcal{V}, \mathcal{E})$ be an ambient graph and $\mathcal{D} = \{(S_i, y_i)\}_{i=1}^{n}$ a labeled training collection of subgraphs $S_i \subseteq G$. A link-prediction stage produces a node table $E_0 \in \mathbb{R}^{|\mathcal{V}| \times d}$ containing global structural context. The downstream model f_θ indexes rows of a trainable copy E_θ when encoding each S_i, with θ collecting the table and predictor parameters. Conventional fine-tuning solves

$$\min_{\theta} \mathbb{E}_{(S,y)\sim\mathcal{D}} \left[\ell\big(f_\theta(S; E_\theta), y\big) \right], \qquad E_\theta \leftarrow E_0 \text{ at initialization.} \tag{1}$$

Shared parameters, subgraph supervision. The same row of E_θ is used by every subgraph containing that node. Consequently, a subgraph label can update many table entries, and one entry can receive updates from several labeled subgraphs. Unlike a collection of separately encoded graphs, these examples do not have independent node parameters: changing a shared row changes the representations available to all subgraphs that query it. The downstream loss reaches that row through the subgraph encoder, aggregation, and classifier. It therefore specifies whether the resulting

subgraph prediction is useful, without assigning a target representation to each participating node. This differs from the pairwise relations used in LP pretraining, even when both stages operate on the same ambient graph.

What complete-view fitting leaves unspecified. The table may contain many more parameters than there are labeled subgraphs. These properties motivate our hypothesis of *underdetermined adaptation*: low training loss alone may leave the table's response to latent perturbations poorly constrained. Two adapted tables and their predictors can fit the observed subgraph labels similarly while using different combinations of pretrained directions. Removing one such combination may have little effect on one predictor but expose substantial classification error in the other. Equation 1 evaluates the complete table and does not directly distinguish these responses. The question is therefore how to guide adaptation toward solutions that retain task information under a specified family of latent transformations, while continuing to optimize complete-view predictions.

This interpretation concerns the constraints supplied by the training objective, rather than parameter count alone. Locally, a table change that leaves the training outputs unchanged to first order lies in the kernel of their Jacobian with respect to the table. Predictions under additional table views can respond to some of those changes and thus provide further constraints; Appendix A formalizes this relationship. Whether these constraints improve downstream classification is assessed by the component and sensitivity experiments.

A view of the shared table. We supplement Equation 1 with supervised loss on partial table views. Each view transforms the current trainable parameters before subgraph aggregation, while retaining the graph, subgraph membership, and training labels. A minibatch thus queries a single transformed ambient table, and the same classifier evaluates both complete and partial views. The labels measure how well task information remains accessible through each view; the partial table is not a reconstruction target. A fixed reference system derived from E_0 keeps the meaning of its directions consistent as E_θ changes.

HardLT requires access to the node table before subgraph aggregation. It applies the same coordinate transformation to every row, making each view *globally coherent*. The complete-view loss directly trains the representation used for standard validation and test prediction, while the selected partial-view loss supplies an additional adaptation signal. The reference basis defines the auxiliary views without restricting the complete table to the PCA subspace. Section 4 specifies the view family and the minibatch-level rule for selecting this signal.

4 HARD LATENT-TABLE ADAPTATION

Framework overview. HardLT uses the complete view to fit the downstream task and a task-selected partial view to constrain adaptation of the shared node table (Figure 1). Link-prediction pretraining supplies E_0, which initializes the trainable table E_θ and defines a fixed PCA mean and basis. The complete-view branch uses E_θ directly. The auxiliary branch projects the current table into this reference system, applies K candidate masks shared across all nodes, and reconstructs the corresponding partial tables in the original feature space.

Roles of the components. The fixed basis defines comparable directions throughout adaptation. A shared mask makes each candidate a single table that all subgraphs in the minibatch query before aggregation. Selection by their mean loss allocates the auxiliary update to a transformation that is difficult for the batch, rather than choosing a different transformation for each example. The complete-view term directly trains the representation used at inference, including components omitted by PCA reconstruction. The table and predictor are updated jointly.

4.1 A FIXED REFERENCE COORDINATE SYSTEM

We center the initial LP table using $\mu_0 = |\mathcal{V}|^{-1} \sum_{v \in \mathcal{V}} E_0[v] \in \mathbb{R}^{1 \times d}$ and fit a rank-r PCA basis $V_0 \in \mathbb{R}^{d \times r}$. This basis is computed once from E_0 and then held fixed; the trainable table E_θ remains free to adapt. At any optimization step we write

$$Z_\theta = (E_\theta - \mathbf{1}\mu_0)V_0, \qquad Z_\theta \in \mathbb{R}^{|\mathcal{V}| \times r}. \tag{2}$$

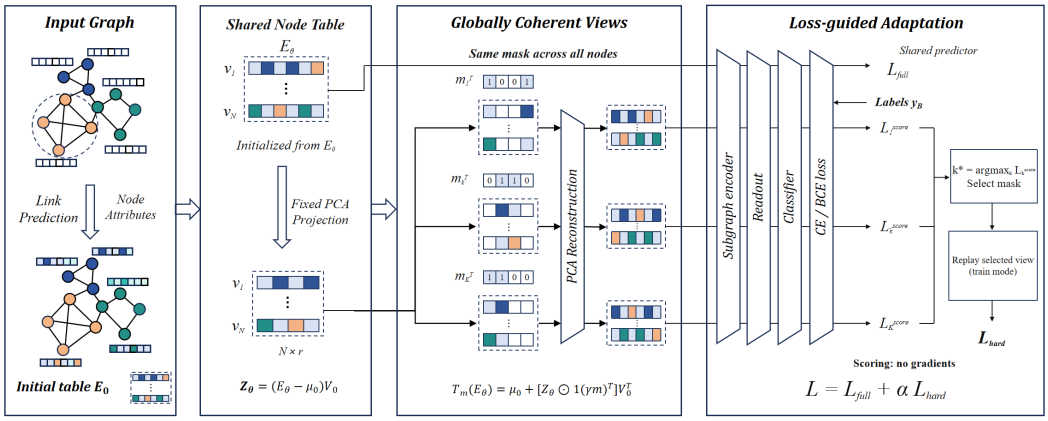

Figure 1: **HardLT framework.** Edge-level link prediction initializes a node table shared by all subgraphs. During adaptation, the complete-view branch fits the downstream task, while the auxiliary branch samples K masks in a fixed PCA coordinate system. Each mask retains the same latent directions across all nodes, producing a globally coherent partial table. A shared predictor scores these views against the subgraph labels without gradients. One view is selected by the minibatch's mean loss and replayed in training mode; its loss augments the complete-view objective. Inference uses only the complete table. Mask patterns and table sizes are schematic. The diagram shows the $\lambda = 0$ objective; configurations with $\lambda > 0$ also use the consistency term in Equation 5.

The fixed basis allows the same latent directions to be perturbed throughout training, even as the node table changes.

4.2 Globally coherent partial table views

A mask $m \in \{0,1\}^r$ retains exactly s of the r coordinates. A fixed protected set P has size q, with $0 \le q \le s \le r$ and $s > 0$. Each mask retains P and samples $s - q$ coordinates uniformly without replacement from the other $r - q$; write this distribution as $\mathcal{M}_{r,s,P}$. The dataset-specific sets are listed in Appendix B. The same m is broadcast to all rows of Z_θ. We reconstruct a partial table as

$$T_m(E_\theta) = \mathbf{1}\mu_0 + \big(Z_\theta \odot \mathbf{1}(\gamma m)^\top\big)V_0^\top, \tag{3}$$

where $\mathbf{1} \in \mathbb{R}^{|\mathcal{V}|}$ broadcasts the mask. We use the common budget scaling $\gamma = r/s$. With uniform sampling over all coordinates and no protected coordinates, this preserves the expected projected coordinates. With protected coordinates, retention probabilities differ and this common scaling is not coordinate-wise unbiased. Rank truncation also prevents an unbiased reconstruction of the complete table in general. Equation 3 reconstructs only the retained PCA component; it does not add back the residual outside the reference basis. Protected coordinates are retained but still multiplied by γ.

Global sharing defines a single transformation of the ambient table. Nodewise masking instead applies different coordinate deletions to different rows, including nodes participating in the same aggregation. Each nodewise mask is reused across occurrences of its row; global masking additionally shares the direction set across distinct rows. Both use the same reference basis. The matched global/nodewise controls isolate this sharing choice.

Effect of sharing the mask. For a fixed linear subgraph aggregation, a shared coordinate perturbation couples the responses of different node rows, whereas independent row masks remove the corresponding cross-row covariance terms. Appendix A.2 derives this difference for linear aggregation; its effect on nonlinear classification is evaluated empirically.

4.3 Hardest-of-K downstream adaptation

For each minibatch B, we draw K masks $m_1, \ldots, m_K \overset{\text{iid}}{\sim} \mathcal{M}_{r,s,P}$. Let ℓ_{ik} denote the classification loss of subgraph i under candidate k, evaluated with the same predictor in evaluation mode (CE, or

BCE averaged over labels). Selection uses the mean over subgraphs:

$$L_k^{\text{score}}(B) = \frac{1}{|B|} \sum_{i \in B} \ell_{ik}, \qquad k^* = \arg\max_k L_k^{\text{score}}(B). \tag{4}$$

The entire minibatch uses m_{k^*}, including every occurrence of a shared node. Masks are resampled at the next minibatch. Keeping the selected mask fixed, we backpropagate

$$L(B) = L_{\text{full}}^{\text{train}}(B) + \alpha L_{\text{selected}}^{\text{train}}(B) + \lambda L_{\text{cons}}(B). \tag{5}$$

The superscripts denote execution modes of the same predictor, not separate models. $L_{\text{full}}^{\text{train}}$ uses the unmodified E_θ, and $L_{\text{selected}}^{\text{train}}$ is the classification loss on the training-mode replay of $T_{m_{k^*}}(E_\theta)$. We do not differentiate through the discrete choice k^*.

Prediction consistency. When $\lambda > 0$, L_{cons} is KL divergence from the detached complete-view prediction to the selected-view prediction, both from their training-mode passes. PPI uses categorical KL and EM uses binary Bernoulli KL, both averaged over examples. HPO-Neuro uses independent Bernoulli KL and averages over both examples and labels. Consistency gradients flow through the partial-view prediction only; the complete branch still receives its supervised gradient. Candidate scoring uses classification loss alone, without the consistency term. Appendix B.1 gives the exact formulas and the averaging rule for multiple views. The main configurations use $\lambda = 0.1$ for PPI-BP and HPO-Neuro, 0.05 for EM-User, and 0 for HPO-Metab. At validation and test time we use $f_\theta(S; E_\theta)$ only. Thus HARDLT adds no inference-time view, ensemble, or parameter.

A common view versus per-example selection. For the same matrix of candidate losses,

$$\max_k \frac{1}{|B|} \sum_{i \in B} \ell_{ik} \;\le\; \frac{1}{|B|} \sum_{i \in B} \max_k \ell_{ik}. \tag{6}$$

This follows by bounding each ℓ_{ik} by its row maximum and then maximizing the left side. HardLT uses the left side to select a common table view. The right side allows different subgraphs to choose different candidates, so their predictions need not correspond to one transformed ambient table. The inequality makes the batch-level selection constraint explicit: the selected mask maximizes aggregate loss, while individual subgraphs can favor different candidates. Batch size and composition therefore affect the selected transformation.

Evaluation-mode scoring suppresses dropout during candidate comparison. Training-mode replay and consistency mean that the update need not equal the gradient of the scoring maximum. For the auxiliary classification term, random $K{=}1$ optimizes an expectation over views and training stochasticity; averaging K losses trains on all sampled views. HardLT instead applies the auxiliary update to one selected view.

Loss-tail weighting. For fixed parameters and deterministic scoring, let $Q_L(u)$ be the quantile function of the scoring loss under one random admissible mask. With K independent candidates, the standard order-statistic identity gives

$$\mathbb{E}[\max_{k \le K} L_k^{\text{score}}] = \int_0^1 Q_L(u)\, K u^{K-1}\, du. \tag{7}$$

Larger K weights the upper scoring-loss tail more heavily, providing a direct interpretation of candidate count as a control on selection pressure. Appendix A.3 derives this scoring identity and distinguishes it from the stochastic training update.

5 EXPERIMENTS

We report four-dataset benchmark results (Table 1), matched HPO-Metab and EM-User component comparisons (Table 2), and HPO-Metab hyperparameter sensitivity (Figure 2), followed by a fixed-view sensitivity evaluation using the same HPO-Metab component-study checkpoints (Figure 3).

Table 1: Mean micro-F1 on four real-world datasets: HardLT reports mean ± standard deviation over 10 runs; baselines are quoted from published results (Wu et al., 2025; Kim & Oh, 2025; Zhong et al., 2025).

Method	PPI-BP	HPO-Metab	HPO-Neuro	EM-User
MLP	0.445 ± 0.009	0.386 ± 0.035	0.404 ± 0.019	0.524 ± 0.060
GBDT	0.446 ± 0.000	0.404 ± 0.000	0.513 ± 0.000	0.694 ± 0.000
Sub2Vec	0.388 ± 0.003	0.472 ± 0.032	0.618 ± 0.009	0.779 ± 0.041
GNN-seg	0.361 ± 0.025	0.542 ± 0.028	0.647 ± 0.003	0.725 ± 0.010
SubGNN	0.599 ± 0.025	0.537 ± 0.025	0.644 ± 0.019	0.816 ± 0.041
GLASS	0.619 ± 0.022	0.614 ± 0.016	0.685 ± 0.016	0.888 ± 0.019
SSNP	0.636 ± 0.022	0.587 ± 0.032	0.682 ± 0.013	0.888 ± 0.016
S2N	0.643 ± 0.041	0.639 ± 0.072	0.686 ± 0.019	0.890 ± 0.035
WLKS-$\{0, D\}$	0.648 ± 0.000	0.579 ± 0.000	0.653 ± 0.000	0.918 ± 0.000
ISNN	$\mathbf{0.731} \pm 0.026$	0.646 ± 0.014	0.688 ± 0.004	0.914 ± 0.009
GPEN	0.644 ± 0.009	0.639 ± 0.009	$\mathbf{0.691} \pm 0.006$	0.912 ± 0.013
HARDLT (ours)	0.648 ± 0.016	$\mathbf{0.647} \pm 0.018$	$\mathbf{0.691} \pm 0.007$	$\mathbf{0.922} \pm 0.013$

Table 2: Component comparisons (ten matched runs, mean ± sample SD). Bold marks the highest mean among the five view objectives, including ties.

Training objective	HPO-Metab	EM-User
Full adaptation	0.634 ± 0.023	0.916 ± 0.024
Global random	0.638 ± 0.016	0.918 ± 0.017
Global average	0.636 ± 0.017	$\mathbf{0.922} \pm 0.019$
Nodewise hard	0.635 ± 0.011	0.908 ± 0.031
HardLT, global hard	$\mathbf{0.647} \pm 0.018$	$\mathbf{0.922} \pm 0.013$

Each task has a fixed ten-seed block; HardLT reuses its main-table runs. Multi-view objectives use $K = 4$ on HPO and $K = 8$ on EM; random uses $K = 1$.

5.1 SETUP AND BENCHMARK COMPARISON

We evaluate PPI-BP, HPO-Metab, HPO-Neuro, and EM-User using the GPEN encoder and data pipeline (Wu et al., 2025). HardLT changes the adaptation objective while retaining this encoder. Scores are micro-F1 and displayed to three decimals. Our errors are sample standard deviations; published errors retain their original definitions. Statistics use unrounded records. Experimental settings are provided in Appendix B.

Benchmark comparison. Among the configurations in Table 1, HardLT achieves the highest mean scores on HPO-Metab, EM-User, and HPO-Neuro, including a tie with GPEN on HPO-Neuro. Compared with published GPEN, its means increase on three datasets and match on HPO-Neuro. The largest gains are on EM-User (0.912 to 0.922) and HPO-Metab (0.639 to 0.647), with the encoder and inference procedure unchanged. The component comparisons below examine this adaptation design under matched configurations.

5.2 COMPONENT ANALYSIS

Table 2 compares complete-view adaptation with four view objectives on HPO-Metab and EM-User. Global random uses one sampled mask; global average averages all candidate losses; nodewise hard replaces the shared mask with independent masks across rows. Within each task, the base configuration and ten seeds are matched, and objectives are compared at the HardLT settings without separate tuning. Both tasks use rank 48, keep 36, and $\alpha = 0.3$; HPO uses $(q, K, \lambda) = (2, 4, 0)$ and EM uses $(0, 8, 0.05)$ for the view objectives. Global average averages the classification and consistency losses separately over its K views, using the same detached full-view target for every

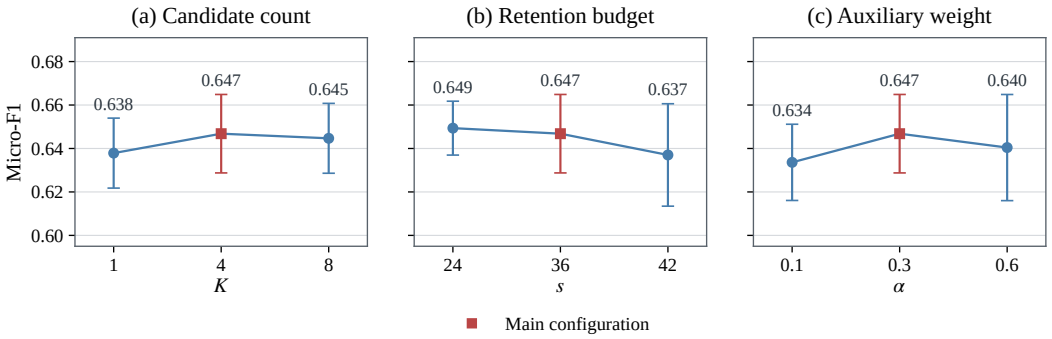

Figure 2: HPO-Metab sensitivity: mean ± sample SD over ten seeds per setting. (a) Candidate count; $K = 1$ is random single-view training. (b) The three evaluated training-retention budgets. (c) Auxiliary weight. Settings are equally spaced within each panel; the center point reuses the main-table HardLT runs.

consistency term. Full adaptation uses only the complete-view classification loss. The HardLT rows reuse the main-table runs.

Effect of partial-view training. On HPO-Metab, where $\lambda = 0$, adding the selected-view supervision raises the mean from 0.634 to 0.647, an increase of 0.013. This comparison shows a higher observed mean from selected-view training without a consistency term. On EM-User, HardLT raises the mean from 0.916 to 0.922, an increase of 0.006. The EM comparison adds both partial-view supervision and consistency ($\lambda = 0.05$), so it evaluates their combined contribution.

Effect of sharing the transformation. To test whether masks should be shared across nodes, we compare HardLT with nodewise hard, which samples masks independently for each node while retaining the same PCA basis, retention budget, candidate count, and minibatch-level hardest-view selection rule. Table 2 shows that sharing the mask raises mean micro-F1 from 0.635 to 0.647 on HPO-Metab and from 0.908 to 0.922 on EM-User. These comparisons support using a common latent-direction transformation across nodes in the evaluated configurations. Appendix A.2 characterizes the resulting cross-row coupling for linear aggregation.

Effect of view selection. Within this fixed-configuration component comparison, hard selection gives the highest HPO-Metab mean, exceeding random and averaged global views by 0.009 and 0.011. On EM-User, HardLT and global averaging both achieve a mean score of 0.922. HardLT uses one selected-view training replay, whereas averaging backpropagates through all eight candidate views. Selection therefore concentrates the auxiliary update on one task-selected transformation; its relative performance against averaging depends on the task.

5.3 Hyperparameter analysis

Figure 2 examines three aspects of the view construction on the same ten HPO-Metab seeds. Each sweep varies one parameter around the fixed main configuration. These are sensitivity measurements, separate from the fixed-configuration objective comparisons in Table 2.

Candidate count K. Moving from one random view to four candidates raises the mean from 0.638 to 0.647. Increasing K further to eight gives 0.645, while requiring more candidate evaluations. Four candidates therefore provide a favorable observed balance between classification performance and scoring cost in this comparison.

Retention budget s. Keeping 24 or 36 of the 48 reference directions gives means of 0.649 and 0.647, compared with 0.637 for 42 directions. Increasing the retention budget does not yield a higher mean here. Because $\gamma = r/s$, this sweep changes both the retained direction count and their scaling. It therefore evaluates the combined view operator at different budgets, while the complete-view branch remains unchanged.

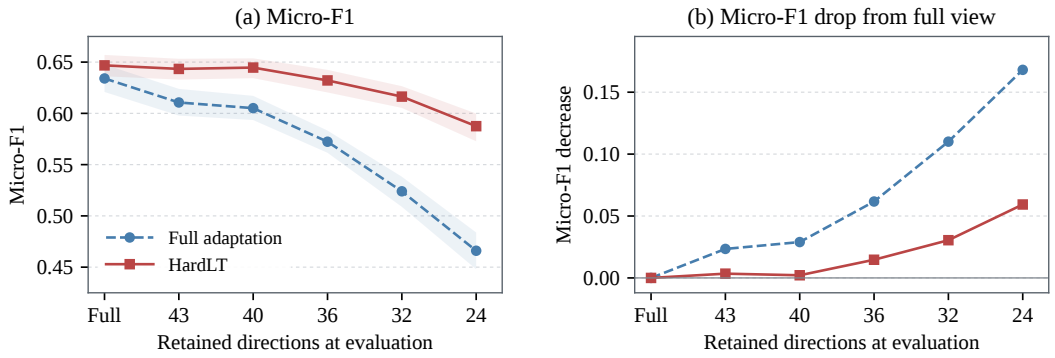

Figure 3: Sensitivity to fixed shared views on HPO-Metab, using the same ten checkpoint pairs as Table 2. (a) Test micro-F1, averaged over 20 masks per budget and then over seeds; shading denotes pointwise bootstrap 95% intervals across seeds. (b) Decrease from each method's complete-view score; lower is better. Full bypasses the transform.

Auxiliary weight α. The intermediate weight 0.3 gives the highest mean, 0.647, compared with 0.634 at 0.1 and 0.640 at 0.6. This nonmonotonic response is consistent with a tradeoff between fitting the complete table and emphasizing the selected partial view. These sweeps describe local sensitivity; the main configuration is not reselected from their test scores.

5.4 Sensitivity to shared representation perturbations

Figure 3 probes sensitivity using the same ten HPO-Metab checkpoint pairs as Table 2. Unlike Figure 2, which varies training settings, it holds the models fixed and varies only the evaluation views. Both methods use the same masks at each retention budget, preserving the two protected coordinates and scaling by $48/s$. Graph structure and subgraph membership remain unchanged.

HardLT has higher mean micro-F1 and a smaller decrease from its own complete-view score at every tested retention budget. At $s = 24$, it achieves 0.588, compared with 0.466 for full adaptation. The corresponding decreases from 0.647 and 0.634 are 0.059 and 0.168. Panel (b) separates sensitivity to the transformations from the initial performance gap: HardLT's advantage persists when measured relative to each method's complete-view score. This supports reduced sensitivity to the evaluated shared transformations, consistent with the motivation for partial-view training.

Training cost and scope. HardLT adds K gradient-free candidate evaluations and a training-mode replay of the selected view to each complete-view update. Inference uses the complete table without additional parameters, view search, or an ensemble. The benchmark study uses the GPEN encoder; the component comparisons examine fixed configurations on HPO-Metab and EM-User.

6 Conclusion

We presented HardLT for adapting LP-pretrained node tables to subgraph tasks. Its central design couples a globally shared latent transformation with minibatch-level loss selection: the complete view fits the task, while the selected view supplies an additional training constraint on the same table. The analysis characterizes how sharing couples node responses and how candidate selection emphasizes the scoring-loss tail. With the GPEN encoder, matched component studies give higher mean scores than complete-view adaptation and nodewise view training on HPO-Metab and EM-User. In the benchmark table, HardLT attains the highest mean scores on HPO-Metab, EM-User, and HPO-Neuro, including a tie with GPEN on HPO-Neuro. Fixed-view evaluation on the same HPO-Metab checkpoints shows smaller performance decreases across the tested retention budgets, supporting the use of shared-view training to reduce sensitivity to these latent transformations. Together, these findings establish shared-view adaptation as a useful training design for the evaluated LP-pretrained subgraph predictor. HardLT trains the shared table under task-selected transformations while preserving the original inference procedure.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the operator and objective; Appendix B provides experimental settings. Tables 1–2 report benchmark and component results with their repetition counts, and Figures 2 and 3 report hyperparameter and fixed-view sensitivity, respectively. Appendix A supplies the assumptions and proofs for the operator analysis.

STATEMENT ON THE USE OF GENERATIVE AI

Generative AI tools supported literature discovery, research ideation, and drafting and polishing parts of the manuscript. They also assisted with mathematical claims and proofs, selected aspects of experimental design and execution, code implementation and debugging, and result analysis and visualization. The authors take responsibility for the final manuscript, methods, and results.

REFERENCES

Carlo Acerbi. Risk aversion and coherent risk measures: A spectral representation theorem. arXiv preprint arXiv:cond-mat/0107190, 2001. URL https://arxiv.org/abs/cond-mat/0107190.

Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B. Aditya Prakash. Sub2Vec: Feature learning for subgraphs. In *Advances in Knowledge Discovery and Data Mining*, pp. 170–182. Springer, 2018. URL https://faculty.cc.gatech.edu/~badityap/papers/sub2vec-pakdd18.pdf.

Emily Alsentzer, Samuel G. Finlayson, Michelle M. Li, and Marinka Zitnik. Subgraph neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/5bca8566db79f3788be9efd96c9ed70d-Abstract.html.

Christopher M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108. URL https://www.microsoft.com/en-us/research/publication/training-with-noise-is-equivalent-to-tikhonov-regularization/.

Jiaao Chen, Dinghan Shen, Weizhu Chen, and Diyi Yang. HiddenCut: Simple data augmentation for natural language understanding with better generalizability. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4380–4390. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.338. URL https://aclanthology.org/2021.acl-long.338/.

Fabio Ferreira, Ivo Rapant, Jörg K. H. Franke, and Frank Hutter. Beyond random augmentations: Pretraining with hard views. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AK1C55o4r7.

Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. MaxUp: Lightweight adversarial training with data augmentation improves neural network training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2474–2483, 2021.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864. ACM, 2016. doi: 10.1145/2939672.2939754. URL https://cs.stanford.edu/~jure/pubs/node2vec-kdd16.pdf.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Shweta Ann Jacob, Paul Louis, and Amirali Salehi-Abari. Stochastic subgraph neighborhood pooling for subgraph classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3963–3967. ACM, 2023. doi: 10.1145/3583780.3615227.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. NEFTune: Noisy embeddings improve instruction finetuning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0bMmZ3fkCk.

Dongkwan Kim and Alice Oh. Translating subgraphs to nodes makes simple GNNs strong and efficient for subgraph representation learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 24415–24436. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24aa.html.

Dongkwan Kim and Alice Oh. Generalizing Weisfeiler–Lehman kernels to subgraphs. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HZgZrtIreg.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2825–2834. PMLR, 2018. URL https://proceedings.mlr.press/v80/li18a.html.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-Drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5a66b9200f29ac3fa0ae244cc2a51b39-Abstract.html.

Mostafa Rahmani and George Atia. Data dropout in arbitrary basis for deep network regularization. *arXiv preprint arXiv:1712.00891*, 2017. URL https://arxiv.org/abs/1712.00891.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Yili Shen, Xiao Liu, Cheng-Wei Ju, Jiaxu Yan, Jun Yi, Zhou Lin, and Hui Guan. Improving subgraph representation learning via multi-view augmentation. *arXiv preprint arXiv:2205.13038*, 2022. URL https://arxiv.org/abs/2205.13038v3.

Suraj Srinivas and R. Venkatesh Babu. Generalized dropout. *arXiv preprint arXiv:1611.06791*, 2016. URL https://arxiv.org/abs/1611.06791.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL https://www.jmlr.org/papers/v15/srivastava14a.html.

Sotaro Takeshita, Yurina Takeshita, Daniel Ruffinelli, and Simone Paolo Ponzetto. Randomly removing 50% of dimensions in text embeddings has minimal impact on retrieval and classification tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 27705–27726. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1410. URL https://aclanthology.org/2025.emnlp-main.1410/.

Stefan Wager, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.nips.cc/paper_files/paper/2013/hash/38db3aed920cf82ab059bfccbd02be6a-Abstract.html.

Xiyuan Wang and Muhan Zhang. GLASS: GNN with labeling tricks for subgraph representation learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XLxhEjKNbXj.

Nannan Wu, Yuming Huang, Yiming Zhao, Jie Chen, and Wenjun Wang. GPEN: Global position encoding network for enhanced subgraph representation learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 67326–67341. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25l.html.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html.

Yongjian Zhong, Liao Zhu, Hieu Vu, and Bijaya Adhikari. Implicit subgraph neural network. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 78602–78615. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhong25g.html.

A Theoretical analysis of shared latent-table views

We analyze the shared-mask distribution, candidate scoring, and the training update following selection. Unless stated otherwise, the graph, reference basis, and labeled batch are fixed. The derivations use standard mask moments, order statistics, and local expansions to characterize the HardLT operator. They provide conditional properties of the training objective, not population generalization or convergence guarantees.

A.1 Exact moments of the implemented view

Let $N = |\mathcal{V}|$, $V \in \mathbb{R}^{d\times r}$ satisfy $V^\top V = I_r$, and let $Z = (E - \mathbf{1}\mu^\top)V$. The fixed mean μ is represented as a column vector in this appendix. For a mask m, write $g = \gamma m$ and

$$T_m(E) = \mathbf{1}\mu^\top + Z\,\mathrm{diag}(g)V^\top.$$

A protected set P of size q is always retained; $t = s - q$ coordinates are sampled uniformly without replacement from the remaining $n = r - q$. Assume $n \geq 2$, $0 \leq t \leq n$, and $s > 0$; write $p = t/n$.

Proposition A.1 (Mean and covariance of an exact-budget mask)**.** *The multiplier mean is $\bar{g}_j = \gamma$ for $j \in P$ and $\bar{g}_j = \gamma p$ otherwise. Protected rows and columns of $C = \mathrm{Cov}(g)$ are zero. Its unprotected block is*

$$C_U = \gamma^2 p(1-p)\frac{n}{n-1}\left(I_n - \frac{\mathbf{1}\mathbf{1}^\top}{n}\right). \tag{8}$$

Consequently, $\mathbb{E}T_m(E) = \mathbf{1}\mu^\top + Z\,\mathrm{diag}(\bar{g})V^\top$.

Proof. Each unprotected indicator has mean t/n and variance $p(1-p)$. For distinct unprotected coordinates, $\mathbb{E}[m_i m_j] = t(t-1)/(n(n-1))$, so $\mathrm{Cov}(m_i, m_j) = -p(1-p)/(n-1)$. Multiplying by γ^2 gives the displayed matrix. Protected indicators are constant, hence have zero covariance. Linearity gives the table mean. □

With $q = 0$ and $\gamma = r/s$, $\bar{g} = \mathbf{1}$, so the mean is the rank-r reconstruction, not necessarily E. Protected coordinates and rank truncation therefore introduce a deterministic mean displacement in addition to random perturbations. Subsequent calculations explicitly retain this displacement instead of assuming an unbiased full-table corruption. If $n \leq 1$, the exact-budget unprotected mask is deterministic; its covariance is zero (an empty block when $n = 0$).

A.2 WHY SHARING CHANGES THE GEOMETRY OF THE CONSTRAINT

For a fixed linear aggregation, let $h_i = \sum_v a_{iv} E[v]^\top$. Write $z_v = Z[v]^\top$, $B_v = V \operatorname{diag}(z_v)$, and $\xi = g - \bar{g}$. The random centered component of a shared-view aggregate is $\delta h_i = \sum_v a_{iv} B_v \xi$.

Proposition A.2 (Cross-row coupling)**.** *For a common multiplier vector across all node rows,*

$$\operatorname{Cov}(\delta h_i, \delta h_j) = \sum_{v,w} a_{iv} a_{jw} B_v C B_w^\top. \tag{9}$$

For independent row multipliers with the same marginal covariance C, where each row's draw is reused across all its occurrences,

$$\operatorname{Cov}(\delta h_i, \delta h_j) = \sum_{v} a_{iv} a_{jv} B_v C B_v^\top. \tag{10}$$

Proof. Expand the covariance of the two linear sums. Under sharing, $\mathbb{E}[\xi\xi^\top] = C$ for every row pair. Under independent row sampling, $\mathbb{E}[\xi_v \xi_w^\top] = 0$ when $v \neq w$, leaving only the diagonal row pairs. □

Thus global sharing preserves cross-row interactions, including interactions between distinct nodes in different subgraphs. It is not equivalent to independent coordinate dropout at each node. Neither covariance uniformly dominates the other. For example, with two scalar row contributions b_1, b_2, unit aggregation weights, and scalar noise variance c, shared variance is $c(b_1 + b_2)^2$ and independent variance is $c(b_1^2 + b_2^2)$. The shared variance is larger for $b_1 = b_2 \neq 0$ and smaller for $b_1 = -b_2 \neq 0$. This example describes projections of centered perturbations, not a one-dimensional exact-budget mask.

Connection to local loss geometry. Let x stack the clean aggregates and let Δ stack their view-induced changes, with mean b and covariance Σ. If ℓ is three times continuously differentiable on all segments from x to $x + \Delta$ and its third derivative has operator norm at most M, Taylor expansion gives

$$\mathbb{E}\ell(x + \Delta) = \ell(x) + \nabla\ell(x)^\top b + \tfrac{1}{2} b^\top H b + \tfrac{1}{2} \operatorname{tr}(H\Sigma) + \mathcal{R}, \quad |\mathcal{R}| \leq \tfrac{M}{6} \mathbb{E}\|\Delta\|^3, \tag{11}$$

where $H = \nabla^2 \ell(x)$. This follows by taking expectations of the pointwise Taylor expansion and using $\mathbb{E}[\Delta\Delta^\top] = \Sigma + bb^\top$. For a quadratic loss the remainder is zero. For a nonlinear network the smoothness assumption is local and need not hold across activation boundaries. Noise-induced regularization has been studied using related local expansions (Bishop, 1995; Wager et al., 2013); here the cross-row terms and nonzero b specify which geometry is being probed. This does not imply a uniformly beneficial regularizer, since curvature, mean displacement, and row interactions all matter.

A.3 HARD SELECTION AS A LOSS-TAIL FUNCTIONAL

Let L be the deterministic evaluation-mode loss of one random admissible mask at fixed parameters. Assume $\mathbb{E}|L| < \infty$; let F and Q be its CDF and generalized quantile. For independent candidates define $R_K = \mathbb{E} \max_{k \leq K} L_k$.

Proposition A.3 (Tail weighting and finite-candidate coverage)**.** *For every integer $K \geq 1$,*

$$R_K = \int_0^1 Q(u) K u^{K-1} \, du, \qquad R_1 \leq R_K \leq R_{K+1}. \tag{12}$$

For a loss threshold a, the chance that scoring encounters a loss above a is $1 - F(a)^K$. For $K > 1$, defining $\operatorname{CVaR}_\tau(L) = (1-\tau)^{-1} \int_\tau^1 Q(u)\, du$,

$$R_K = \int_0^1 K(K-1)\tau^{K-2}(1-\tau) \operatorname{CVaR}_\tau(L) \, d\tau, \tag{13}$$

$$\mathbb{E}L \leq R_K \leq \operatorname{CVaR}_{1-1/K}(L). \tag{14}$$

Proof. The maximum has CDF $F(a)^K$. Representing each loss as $Q(U_k)$ with independent uniform U_k shows that their maximum is distributed as $Q(U_{(K)})$, whose uniform order statistic has density Ku^{K-1}. Coupling the first K draws with a $(K+1)$st proves monotonicity. Substituting the quantile definition into Equation 13 and exchanging integrals gives $\int_0^1 Q(u) \int_0^u K(K-1)\tau^{K-2} d\tau \, du = R_K$. Absolute integrability justifies this exchange. The mixing weight is nonnegative and integrates to one. For the upper bound, set $w(u) = Ku^{K-1}$ and $w_*(u) = K\mathbf{1}\{u > 1 - 1/K\}$. Both integrate to one. Below $u_0 = 1 - 1/K$, $(Q(u) - Q(u_0))(w(u) - w_*(u)) \le 0$; above u_0 the same inequality holds because $w(u) \le K$. Integrating proves $\int Qw \le \int Qw_*$, including quantile jumps. □

Quantile-weighted loss functionals belong to the spectral-risk framework (Acerbi, 2001). Equation 13 characterizes the sampled scoring maximum as a mixture of tail levels, rather than CVaR at one prescribed level. Increasing K increases tail emphasis at fixed parameters; it does not imply that optimizing with a larger K improves clean accuracy. Likewise, high-probability coverage of a mask-loss tail concerns the specified mask distribution, not arbitrary deployment shifts.

A.4 What additional views can constrain

Let $e = \mathrm{vec}(E) \in \mathbb{R}^{Nd}$, hold the other model parameters fixed, and let $\Phi_0(e) \in \mathbb{R}^a$ stack clean training logits. Let Φ_j stack logits for a fixed view j. At a differentiable point write $J_j = D\Phi_j(e)$ and let J_* vertically stack $J_0, \ldots, J_M$.

Proposition A.4 (First-order invisible directions)**.** *The first-order invisible table perturbations for the clean output form* $\ker J_0$*, of dimension* $Nd - \mathrm{rank}(J_0)$*. For clean and all specified view outputs together, they form*

$$\ker J_* = \bigcap_{j=0}^{M} \ker J_j \subseteq \ker J_0. \tag{15}$$

The inclusion is strict exactly when some $v \in \ker J_0$ *satisfies* $J_j v \neq 0$ *for at least one specified view.*

Proof. Differentiability gives $\Phi_j(e + \varepsilon v) - \Phi_j(e) = \varepsilon J_j v + o(\varepsilon)$. Every first-order change vanishes precisely when every $J_j v = 0$. The dimension statement is rank–nullity; the strict-inclusion criterion follows directly from intersection of kernels. □

This is a local output-observability statement, not proof that training labels identify the table or that the optimizer uses every additional constraint. In particular, output changes can themselves be invisible to a loss, and the actual method selects only sampled views.

A preserved ambiguity. For a fixed linear pooling matrix A, every ΔE with $A\Delta E = 0$ satisfies

$$A\big(T_m(E + \Delta E) - T_m(E)\big) = A\Delta E V \,\mathrm{diag}(g) V^\top = 0 \tag{16}$$

for every shared mask. The fixed reference mean cancels in the difference. Thus shared linear views cannot remove this pooling nullspace, even with unlimited mask coverage. The motivating granularity gap is compatible with additional sensitivity constraints without implying complete identifiability.

A.5 Scoring versus stochastic replay

Fix a candidate set and current parameters. Let S_k be evaluation-mode scoring classification loss and R_k the expected training-mode classification loss when candidate k is replayed from the same model state. The expectation is over replay randomness. The implementation chooses $k^* \in \arg\max_k S_k$.

Proposition A.5 (Conditional replay gap)**.** *If* $|R_k - S_k| \le \varepsilon$ *for all candidates, then*

$$0 \le \max_k R_k - R_{k^*} \le 2\varepsilon. \tag{17}$$

For an idealized deterministic replay with differentiable scores and a unique maximizing candidate, differentiating the selected score equals differentiating the finite maximum locally.

Proof. Let $j \in \arg\max_k R_k$. Then $R_j \le S_j + \varepsilon \le S_{k^*} + \varepsilon \le R_{k^*} + 2\varepsilon$. For the second assertion, the finite strict gap at the unique maximizer persists in a neighborhood by continuity, so the maximum there is the same differentiable score function. □

The bound is conditional; we do not assume or empirically certify a small ε for every training step. It bounds loss values, not gradients. Training-mode stochasticity, including dropout, can invalidate an identification of the actual update with the gradient of R_K. This distinction is why the scoring analysis is not asserted as a convergence theorem for HardLT.

A.6 Full-view fitting and selection among comparable fits

Consider an idealized objective $J_\alpha(\theta) = F(\theta) + \alpha R(\theta)$, where F is complete-view empirical risk, $R \geq 0$ is a specified view risk, and $\alpha > 0$. These functions need not be convex. This abstraction describes the supervised two-term objective with $\lambda = 0$; it does not model the detached-target consistency update used when $\lambda > 0$.

Proposition A.6 (Empirical fit–view-risk tradeoff)**.** *Suppose θ_α globally minimizes J_α and θ_0 globally minimizes F, with finite risks. Then*

$$0 \leq F(\theta_\alpha) - F(\theta_0) \leq \alpha\big(R(\theta_0) - R(\theta_\alpha)\big) \leq \alpha R(\theta_0). \tag{18}$$

For any two candidates a, b with equal F, $J_\alpha(a) < J_\alpha(b)$ if and only if $R(a) < R(b)$. More generally, if $F(a) - F(b) \leq \eta$ and $R(b) - R(a) > \eta/\alpha$, then $J_\alpha(a) < J_\alpha(b)$.

Proof. Global optimality implies $F(\theta_\alpha) + \alpha R(\theta_\alpha) \leq F(\theta_0) + \alpha R(\theta_0)$. Rearrange, use optimality of θ_0 for the lower bound, and use $R \geq 0$ for the last inequality. The remaining claims follow by subtracting the two objective values. □

This elementary comparison formalizes the role of the full-view term: view risk can distinguish equal or sufficiently close empirical fits, while its weight controls a possible cost in complete-view fit. It does not assert that actual SGD reaches a global minimizer, that two trained models have equal training risk, or that smaller view risk guarantees better test performance. The results explain how shared masks couple node perturbations and how candidate selection emphasizes high-loss views; the experiments assess whether these properties improve classification.

B Experimental settings

The primary rank/keep/q/K/α/λ settings for PPI, Metab, Neuro, and EM are respectively $(48, 36, 5, 4, .3, .1)$, $(48, 36, 2, 4, .3, 0)$, $(36, 32, 1, 2, .3, .1)$, and $(48, 36, 0, 8, .3, .05)$. The PPI-BP main-table entry uses the initial ten-run block with seeds $2^j - 1$, $j = 0, \ldots, 9$. The HPO-Metab component-study seeds are $2^j - 1$, $j = 23, \ldots, 32$.

Reference tables and masks. The loaded pretrained table widths d are 72, 64, 74, and 88 for PPI-BP, HPO-Metab, HPO-Neuro, and EM-User, respectively. PCA is fitted to all rows of each initial table and ordered by decreasing variance. With one-based coordinate indices, the fixed protected sets are $P = \{1, 2, 3, 4, 5\}$, $\{1, 3\}$, $\{1\}$, and $\varnothing$, respectively. The nonempty sets come from the initialization-time leave-one-direction-out edge-prediction loss ranking; they remain fixed throughout downstream training. Protection specifies the sampling family, not a guarantee that source predictions are preserved. The original table width is retained after reconstruction and at inference.

Training configuration. We retain the GPEN data pipeline, including its training-subgraph augmentation, and jointly optimize the table and predictor with Adam (Kingma & Ba, 2015). In the same dataset order, batch sizes are $(97, 54, 109, 8)$ and initial learning rates are $(.0023, .0007, .0035, .0013)$. Dropout is 0.5. The reference mean and basis remain fixed. Cross-entropy is used for single-label multiclass targets and binary cross-entropy for binary or multi-label targets. Multiclass predictions use the largest logit; binary and multi-label predictions threshold logits at zero.

B.1 Consistency loss and multi-view objectives

Let $n = |B|$, and let $z^{\mathrm{f}}, z^{(k)}$ be the logits of the complete-view training pass and candidate k's training replay. For PPI-BP's C-class task, use $p = \mathrm{stopgrad}(\mathrm{softmax}(z^{\mathrm{f}}))$ and $q^{(k)} = \mathrm{softmax}(z^{(k)})$. EM-User uses one binary logit per example: with $u_i = \sigma(z_i^{\mathrm{f}})$ and $v_i^{(k)} = \sigma(z_i^{(k)})$, form $p_i =$

$\mathrm{stopgrad}[(1-u_i, u_i)]$ and $q_i^{(k)} = (1-v_i^{(k)}, v_i^{(k)})$, taking $C = 2$. Here σ is the logistic sigmoid. In both cases,

$$D_B(p\|q^{(k)}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} p_{ic}\log\frac{p_{ic}}{q_{ic}^{(k)}}. \tag{19}$$

Thus classes are summed, not averaged. For HPO-Neuro's multi-label task with C labels, set $p = \mathrm{stopgrad}(\sigma(z^{\mathrm{f}}))$ and $q^{(k)} = \sigma(z^{(k)})$, and use

$$D_B(p\|q^{(k)}) = \frac{1}{nC}\sum_{i=1}^{n}\sum_{c=1}^{C}\left[p_{ic}\log\frac{p_{ic}}{q_{ic}^{(k)}} + (1-p_{ic})\log\frac{1-p_{ic}}{1-q_{ic}^{(k)}}\right]. \tag{20}$$

This averages independent Bernoulli divergences over both batch and label dimensions, matching the reduction of the supervised BCE loss. All nonzero-consistency configurations reported here use these KL forms, with no temperature scaling. The complete-view target is detached; only the view-prediction path contributes consistency gradients to the shared table and predictor. Both logits come from training-mode passes, rather than the evaluation-mode candidate scores.

For a set of replayed candidates $\mathcal{S}$, the implemented objective is

$$L(B) = L_{\mathrm{full}}^{\mathrm{train}}(B) + \frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}}\left[\alpha L_k^{\mathrm{train}}(B) + \lambda D_B(p\|q^{(k)})\right]. \tag{21}$$

HardLT and nodewise hard use $\mathcal{S} = \{k^*\}$; random single-view training uses its one sampled candidate. Global average uses all K candidates, averaging their supervised losses and KL terms separately. Every KL term uses the same detached complete-view target from that minibatch. The implementation does not first average probabilities or logits and then compute KL. For HardLT, $L_{\mathrm{cons}}(B) = D_B(p\|q^{(k^*)})$. Full adaptation omits both auxiliary terms.

Statistical reporting. We report means and sample SDs across seeds, with run counts in captions. Comparisons describe observed means.

Fixed-view evaluation. Figure 3 reuses the HPO-Metab full-adaptation and HardLT checkpoints from Table 2, without retraining or checkpoint reselection. This is a post-hoc test sensitivity analysis. At each of the five retention budgets, both methods use the same bank of 20 fixed masks. We first average micro-F1 over masks within a checkpoint, then average over ten seeds. Pointwise 95% percentile bootstrap intervals use 50,000 resamples of the ten seed-level means; masks are not treated as independent runs. Complete-view evaluation bypasses PCA reconstruction.