ADAPTIVE MASKED RECONSTRUCTION FOR MULTI-SCALE CIRCRNA-MIRNA REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting circRNA-miRNA interactions requires learning from incomplete association networks and RNA sequence-derived evidence. We present SMASH, a framework that combines multi-scale hypergraph learning with adaptive masked reconstruction. SMASH represents each RNA using its training association profile and fused sequence, structural, and composition similarities. These features are propagated through interaction-derived and similarity-derived structures. The model retains representations from successive propagation depths and learns RNA-specific weights to combine them, allowing each RNA to draw on different neighborhood ranges. To provide auxiliary supervision, a reconstruction branch recovers masked RNA representations. Its masking probabilities are adjusted using historical reconstruction errors, assigning higher probabilities to RNAs with larger errors while maintaining the average masking probability. The encoder is trained jointly on interaction prediction and masked reconstruction. The resulting RNA representations are then used by a CatBoost classifier to score candidate interactions. Under association-level five-fold cross-validation, SMASH achieves mean AUC/AUPR values of 0.9470/0.9487, 0.9604/0.9596, and 0.9572/0.9566 on CMI-9905, CMI-9589, and CMI-20208, respectively.

1 INTRODUCTION

CircRNA can function as a molecular sponge, sequestering miRNA molecules and preventing their interaction with target messenger RNA (mRNA) (Ma et al., 2024). This mechanism indirectly regulates target gene expression and influences multiple cellular processes involved in cancer progression. For example, in the progression of hepatocellular carcinoma, circ-ZEB1 promotes the expression of PIK3CA by silencing miR-199a-3p, thereby affecting the proliferation and apoptosis of hepatocellular carcinoma cells (Liu et al., 2022). In non-small cell lung cancer (NSCLC), circ-SLC16A1 regulates miR-1287-5p, thereby influencing tumor cell migration and proliferation (Jin et al., 2024). These studies motivate further investigation of circRNA-miRNA interactions and their potential relevance to the diagnosis, treatment, and prognosis of complex human diseases.

Accumulating biological evidence indicates that circRNA-miRNA interactions (CMIs) are involved in cancer progression and may affect therapeutic outcomes (Pisignano et al., 2023; Zhang et al., 2023). However, experimental validation of CMIs is time-consuming and costly, and it requires specialized technical expertise (Dong et al., 2013). Advances in computational methods have led to the development of numerous approaches for predicting circRNA-miRNA interactions (Jiang et al., 2026). For example, Wei et al. (2025) developed EGATCMI, a multi-feature fusion prediction model integrating Transformer architecture and graph attention networks. This model employs Transformer and Word2vec algorithms for RNA sequence pre-training to capture sequence feature representations and similarities. Yin et al. (2024) proposed MRHRL, which models high-order RNA associations through multi-relational hypergraph representation learning and combines multiple relation views using attention.

Graph-based frameworks represent biological entities as nodes and their interactions as edges, enabling neighborhood information to propagate through the resulting topology (Li et al., 2023; Krishnan & Ghebrehiwet, 2024; Wang et al., 2025). However, numerous existing prediction approaches

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.

still incorporate structural prior knowledge in a rigid and static manner (Cheng et al., 2008). Fixed structural priors may propagate errors when the observed interaction network is sparse or contains uncertain associations (Amaro & Mulholland, 2018). Some hypergraph models use a fixed propagation depth. This design applies the same receptive field to all nodes and omits depth-specific intermediate representations (Zahradník et al., 2023; Yu et al., 2023). These observations motivate RNA-specific aggregation across propagation depths and adaptive construction of propagation structures.

SMASH combines interaction profiles with fused sequence, structural, and composition similarities for sparse CMI prediction. The similarity component provides biological evidence independent of the annotated interaction labels. Multi-scale hypergraph propagation produces representations at successive depths, and node-specific weights combine these representations into the backbone output. An auxiliary reconstruction branch masks selected backbone representations before encoding. After a uniform-sampling warm-up, node selection probabilities are adjusted using historical reconstruction errors. The mean masking probability remains fixed. It then zeros the corresponding encoder outputs before hypergraph decoding. The reconstruction targets are the backbone representations computed before masking. This auxiliary objective is optimized jointly with supervised association prediction. Figure 1 summarizes the complete prediction pipeline.

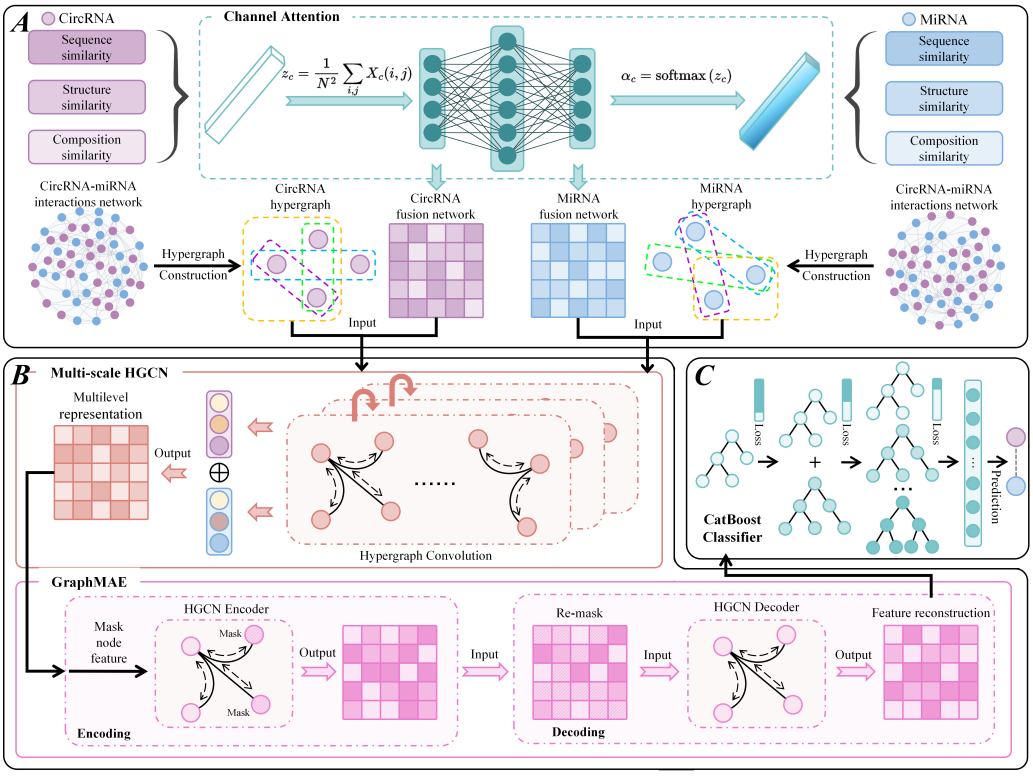

Figure 1: Overview of SMASH. (A) The molecular similarity network is computed and integrated via channel attention, then combined with the corresponding circRNA and miRNA hypergraphs. The connected circles schematically depict parameter-free softmax normalization. (B) Multi-scale HGCN is employed to capture multi-level relationships, while GraphMAE reconstructs masked backbone representations. (C) Potential CMIs are predicted using the CatBoost classification algorithm.

2 Related Work

2.1 Computational prediction of circRNA–miRNA interactions

CircRNA–miRNA interaction prediction seeks to identify potential regulatory associations between circRNAs and miRNAs, reducing the cost of experimental validation. Early methods mainly relied

on sequence complementarity, molecular similarity, or handcrafted network features. Recent deep learning methods have improved CMI prediction by exploiting heterogeneous biological information. BioDGW-CMI (Wang et al., 2023) introduces biological text mining and wavelet diffusion-based network embedding. BEROLECMI (Wang et al., 2024) models role-aware molecular attributes from similarity and biological networks. EGATCMI (Wei et al., 2025) combines sequence representation with graph attention learning, while MRHRL (Yin et al., 2024) captures high-order RNA associations through multi-relational hypergraph representation learning. Despite their effectiveness, these methods often rely on static interaction structures, which may propagate noisy topological priors and restrict adaptive association modeling.

2.2 Graph and hypergraph learning for biological networks

Graph neural networks model molecular interactions by propagating information over graph topology (Rochman Sharabi & Louppe, 2023). Conventional graphs encode pairwise relations. Hypergraphs represent group-wise dependencies through hyperedges. Existing hypergraph-based and multi-hop CMI models can aggregate high-order neighborhoods. Their final representations commonly depend on a fixed propagation depth or merge intermediate outputs without RNA-specific scale selection (Zahradník et al., 2023; Yu et al., 2023). Consequently, the same receptive field is applied to RNAs whose useful structural context may occur at different propagation ranges.

2.3 Position of this work

Channel attention, hypergraph convolution, and masked autoencoding are established operators. SMASH contributes a task-specific formulation that assigns distinct roles to observed interactions and label-independent similarities. Fold-specific RNA-centered hyperedges encode interaction groups, and RNA-specific propagation matrices are extracted from the joint normalized operator. Similarity-derived features provide RNA evidence for propagation and masked reconstruction. Node-specific scale weights select among the retained propagation depths. This organization specifies the representation-learning pipeline for sparse CMI prediction.

Adaptive masking has been studied using learned sampling distributions in video autoencoders (Bandara et al., 2023) and structure-aware curricula for graph reconstruction (Li et al., 2025). Here, we specify a node-level feedback rule based on detached historical reconstruction errors.

3 Method

3.1 Construction of multi-modal similarity

We construct three within-type similarity matrices for circRNAs and miRNAs from sequence, predicted secondary structure, and dinucleotide composition. These matrices encode sequence similarity, structural similarity, and dinucleotide-composition similarity, respectively.

Sequence Similarity: Sequence similarity is computed separately within circRNAs and within miRNAs. Let $s_{r,i}$ denote the nucleotide sequence of RNA i of type $r \in \{c, m\}$. We define $D_{\mathrm{Lev}}(s_{r,i}, s_{r,j})$ as the minimum number of single-nucleotide insertions, deletions, and substitutions required to transform one sequence into the other, assigning unit cost to each operation. For nonempty sequences, the similarity score is

$$S_r^{\mathrm{seq}}(i,j) = 1 - \frac{D_{\mathrm{Lev}}(s_{r,i}, s_{r,j})}{|s_{r,i}| + |s_{r,j}|}. \tag{1}$$

Larger scores indicate greater sequence similarity under this edit-based measure. The resulting matrices provide within-type sequence features for similarity fusion.

Structural Similarity: We compute structural similarity separately for circRNAs and miRNAs. For each RNA i of type $r \in \{c, m\}$, we use RNAfold to predict a minimum free energy secondary structure under the selected folding model. Let $b_{r,i}$ denote its dot-bracket representation, in which matched parentheses identify paired nucleotides and dots indicate unpaired nucleotides. The structural similarity is

$$S_r^{\mathrm{str}}(i,j) = \frac{2M_r(i,j)}{q_{r,i} + q_{r,j}}, \qquad r \in \{c, m\}, \tag{2}$$

The resulting matrices $S_c^{\mathrm{str}} \in \mathbb{R}^{n_c \times n_c}$ and $S_m^{\mathrm{str}} \in \mathbb{R}^{n_m \times n_m}$ provide the structural similarity views for channel attention.

Composition Similarity: For each RNA, we compute a normalized frequency vector $\mathbf{f}_{r,i} \in \mathbb{R}^{16}$ from the 16 dinucleotide combinations of adenine, cytosine, guanine, and uracil. Here, $r \in \{c, m\}$ denotes the RNA type and i indexes an RNA of that type. Composition similarity is the Spearman rank correlation between two such vectors:

$$S_r^{\mathrm{com}}(i,j) = \rho_{\mathrm{S}}(\mathbf{f}_{r,i}, \mathbf{f}_{r,j}), \qquad r \in \{c, m\}. \tag{3}$$

Here, ρ_{S} denotes Spearman rank correlation. The matrices S_c^{com} and S_m^{com} describe dinucleotide-composition similarity within circRNAs and miRNAs, respectively.

3.2 Parameter-Free Channel Attention

For each RNA type $r \in \{c, m\}$, we use parameter-free channel attention to fuse the sequence, structural, and composition similarity matrices (Lin et al., 2024; Zhang & Sivanandan, 2025). Let $\{S_r^k\}_{k=1}^3$, $S_r^k \in \mathbb{R}^{n_r \times n_r}$, denote the sequence, structural, and composition views in this order, with $S_r^3 = S_r^{\mathrm{com}}$. Global average pooling summarizes channel k as

$$z_r^k = \frac{1}{n_r^2} \sum_{i=1}^{n_r} \sum_{j=1}^{n_r} S_r^k(i,j). \tag{4}$$

The channel coefficients and fused similarity matrix are computed as

$$\alpha_r^k = \frac{\exp(z_r^k)}{\sum_{q=1}^3 \exp(z_r^q)}, \qquad \widetilde{S}_r = \sum_{k=1}^3 \alpha_r^k S_r^k. \tag{5}$$

The coefficients satisfy $\sum_{k=1}^3 \alpha_r^k = 1$. Each coefficient is a deterministic function of the input similarities. This calculation contains no trainable weights or biases. The dense connections in Figure 1A illustrate the shared normalization across channels: each coefficient depends on every pooled channel descriptor through the softmax denominator. The groups of circles in Figure 1A are schematic representations of intermediate normalization quantities. The implemented mapping is $\boldsymbol{\alpha}_r = \mathrm{softmax}(\mathbf{z}_r)$, applied directly to the pooled channel descriptors.

3.3 Multi-Scale Hypergraph Convolutional Network

Let $Y^{(f)} \in \{0,1\}^{n_c \times n_m}$ denote the training CMI matrix for fold f. Its nonzero entries identify observed training interactions. We construct a joint hypergraph with circRNAs followed by miRNAs in the node ordering. Each RNA defines a hyperedge containing itself and its interaction partners in $Y^{(f)}$. With hyperedges ordered by their centers, the incidence matrix is

$$\mathcal{H}^{(f)} = \begin{bmatrix} I_{n_c} & Y^{(f)} \\ (Y^{(f)})^\top & I_{n_m} \end{bmatrix}. \tag{6}$$

For unit hyperedge weights, define vertex and hyperedge degrees by

$$d_v(i) = \sum_e \mathcal{H}_{ie}^{(f)}, \qquad d_e(e) = \sum_i \mathcal{H}_{ie}^{(f)}. \tag{7}$$

We use the inverse-degree convention

$$q_\alpha(d) = \begin{cases} d^{-\alpha}, & d > 0, \\ 0, & d = 0, \end{cases} \quad Q_v = \mathrm{diag}(q_{1/2}(d_v(i))), \quad Q_e = \mathrm{diag}(q_1(d_e(e))). \tag{8}$$

The joint normalized operator and its RNA-specific principal submatrices are

$$\begin{aligned} \mathcal{G}^{(f)} &= Q_v \mathcal{H}^{(f)} Q_e (\mathcal{H}^{(f)})^\top Q_v, \\ G_c^{\mathrm{init},(f)} &= \mathcal{G}^{(f)}[\mathcal{C}, \mathcal{C}], \qquad G_m^{\mathrm{init},(f)} = \mathcal{G}^{(f)}[\mathcal{M}, \mathcal{M}]. \end{aligned} \tag{9}$$

Center membership guarantees positive vertex and hyperedge degrees in this initial construction, including for RNAs without training interactions. An empty hyperedge contributes zero under the inverse-degree convention; none arise in Eq. (6). Self-contributions follow from center membership, with no additional identity matrix added after normalization.

The backbone inputs concatenate association profiles and fused similarities:

$$X_c = [Y^{(f)} \| \widetilde{S}_c], \qquad X_m = [(Y^{(f)})^\top \| \widetilde{S}_m]. \tag{10}$$

Here, $\|$ denotes column-wise concatenation, giving $n_c + n_m$ features per node. The incidence matrix, degree factors, and initial propagation matrices are rebuilt for each fold. Association profiles use the same $Y^{(f)}$. We suppress the fold index below.

Stacking hypergraph convolution layers extends node–hyperedge–node propagation. Multi-scale HGCN retains the output at each depth and learns node-specific aggregation weights, allowing each RNA to emphasize a different propagation range.

At each forward pass, the adaptive module maps complete fused similarity profiles $R_r = \widetilde{S}_r$ to learned representations. Their inner products determine a binary incidence matrix B_r; learned coefficients $\boldsymbol{\xi}_r$ scale its rows. Appendix A.1 defines the input, rank selection, and parameters. With $\widehat{B}_r = \mathrm{diag}(\boldsymbol{\xi}_r) B_r$, the backbone and reconstruction branch use

$$G_r^{\mathrm{adaptive}} = Q_{v,r} \widehat{B}_r Q_{e,r} \widehat{B}_r^\top Q_{v,r}, \qquad \bar{G}_r = 0.7 G_r^{\mathrm{init}} + 0.3 G_r^{\mathrm{adaptive}}. \tag{11}$$

Here, $Q_{v,r}$ and $Q_{e,r}$ apply the inverse-degree convention above to the row sums and column sums of $\widehat{B}_r$, with exponents $1/2$ and 1, respectively.

For RNA type r, we define the hypergraph convolution operator as an affine feature transformation followed by propagation through $\bar{G}_r$:

$$\mathrm{HGCN}_s(X, \bar{G}_r) = \bar{G}_r \left(X W^s + \mathbf{1} (b^s)^\top \right). \tag{12}$$

Here, W^s and b^s are trainable parameters. We omit the RNA-type index on the layer parameters and depth-specific representations for clarity. Starting from $X^0 = X_{\mathrm{input}} = X_r$, the representation at depth $s \in \{1, \ldots, N_{\mathrm{scales}}\}$ is computed as

$$X^s = \mathrm{Dropout}_p \left(\mathrm{LeakyReLU}_a \left(\mathrm{HGCN}_s(X^{s-1}, \bar{G}_r) \right) \right), \tag{13}$$

where $a = 0.25$ is the negative slope and $p = 0.2$ is the dropout rate. Each propagation layer applies one LeakyReLU activation. Dropout is enabled during training. Each resulting representation is retained for scale aggregation and supplied to the next propagation layer when one follows.

To effectively integrate multi-scale representations, an attention mechanism was adopted. For each node v, this mechanism computes a set of scale-specific attention scores $A_{v,s}$, by first applying a shared linear transformation to the node features from each scale $X^s[v,:]$, followed by a ReLU activation, and then another linear transformation:

$$A_{v,s} = \mathrm{Linear}_2 \left(\mathrm{ReLU} \left(\mathrm{Linear}_1 \left(X^s[v,:] \right) \right) \right) \tag{14}$$

Softmax converts these scores into node-specific weights across scales:

$$\widetilde{A}_{v,s} = \frac{\exp(A_{v,s})}{\sum_{k=1}^{N_{\mathrm{scales}}} \exp(A_{v,k})}. \tag{15}$$

The scale-fused representation is their weighted sum:

$$X_{\mathrm{fused}}[v,:] = \sum_{s=1}^{N_{\mathrm{scales}}} \widetilde{A}_{v,s} X^s[v,:]. \tag{16}$$

For RNA type r, the residual path uses the backbone input X_r, which concatenates association profiles and fused similarities. The notation X_{input} above refers to this type-specific input. Let $d = 128$ be the width of the retained representations. The residual map is

$$P_r^{\mathrm{res}}(X_r) = \begin{cases} X_r, & \text{if the input width is } d, \\ X_r W_r^{\mathrm{res}} + \mathbf{1} (b_r^{\mathrm{res}})^\top, & \text{otherwise.} \end{cases} \tag{17}$$

The final backbone representation is

$$U_r = \mathrm{LeakyReLU}\left(P_r^{\mathrm{out}}\left(X_{r,\mathrm{fused}} + P_r^{\mathrm{res}}(X_r)\right)\right) \in \mathbb{R}^{n_r \times 128}, \tag{18}$$

where $X_{r,\mathrm{fused}}$ denotes the scale-fused representation for RNA type r. The output map P_r^{out} is a trainable affine map with its own parameters. Both projections and the scale-attention parameters are learned jointly with the backbone. The complete representation U_r, including the residual contribution, is computed before masking.

3.4 Adaptive Masked Reconstruction of Backbone Representations

We adopt input masking and re-mask decoding inspired by GraphMAE (Hou et al., 2022) to regularize the backbone representations. The auxiliary branch receives U_r, obtained after scale aggregation, residual addition, output projection, and activation. Its reconstruction target is the corresponding backbone representation computed before masking. We specify a feedback-based sampler that adjusts node selection probabilities during training. Each RNA type maintains a separate error state $d_{r,i}^{(0)} = 1$, reset at the start of each fold. At iteration t, define the centered historical errors by

$$\bar{d}_r^{(t-1)} = \frac{1}{n_r}\sum_i d_{r,i}^{(t-1)}, \qquad v_{r,i}^{(t)} = d_{r,i}^{(t-1)} - \bar{d}_r^{(t-1)}. \tag{19}$$

The node selection probabilities are

$$p_{r,i}^{(t)} = \rho + \eta_t \min(\rho, 1-\rho)\frac{v_{r,i}^{(t)}}{\max_j |v_{r,j}^{(t)}| + \epsilon}, \qquad \eta_t = \begin{cases} 0, & t \le T_0, \\ \eta, & t > T_0. \end{cases} \tag{20}$$

We set $\rho = 0.5$, $T_0 = 5$, $\eta = 0.5$, and $\epsilon = 10^{-8}$. The first T_0 iterations use uniform sampling. Subsequently, nodes with larger historical reconstruction errors receive higher selection probabilities. The centering gives $n_r^{-1}\sum_i p_{r,i}^{(t)} = \rho$. For these settings, $p_{r,i}^{(t)} \in [0.25, 0.75]$. Equal error states recover uniform sampling. Each node is sampled independently:

$$m_{r,i}^{(t)} \sim \mathrm{Bernoulli}(p_{r,i}^{(t)}), \qquad \Omega_r^{(t)} = \{i : m_{r,i}^{(t)} = 1\}. \tag{21}$$

Thus, ρn_r is the expected number of selected nodes; the realized count can vary. We suppress t in the reconstruction equations below. The entire representation of each selected node is replaced by a learnable token:

$$\widetilde{U}_r[i,:] = \begin{cases} \mathbf{t}_r, & i \in \Omega_r, \\ U_r[i,:], & i \notin \Omega_r. \end{cases} \tag{22}$$

Each RNA type has its own learnable input mask token and auxiliary encoder–decoder parameters. A single hypergraph convolutional encoder maps the masked backbone representations to

$$H_r^{\mathrm{enc}} = \bar{G}_r\left(\widetilde{U}_r W_r^{\mathrm{enc}} + \mathbf{1}(b_r^{\mathrm{enc}})^\top\right). \tag{23}$$

Before decoding, we re-mask the encoder outputs at the same node indices selected for input masking. The corresponding rows are replaced by fixed zero vectors:

$$\widetilde{H}_r^{\mathrm{enc}}[i,:] = \begin{cases} \mathbf{0}, & i \in \Omega_r, \\ H_r^{\mathrm{enc}}[i,:], & i \notin \Omega_r. \end{cases} \tag{24}$$

The hypergraph decoder reconstructs the backbone representations from the re-masked encoder outputs:

$$\widehat{U}_r = \bar{G}_r\left(\widetilde{H}_r^{\mathrm{enc}} W_r^{\mathrm{dec}} + \mathbf{1}(b_r^{\mathrm{dec}})^\top\right). \tag{25}$$

Both masking operations use the same node set Ω_r. Re-masking removes the selected nodes' own encoder outputs from the decoder input. The decoder aggregates the retained encoder representations through $\bar{G}_r$. The reconstruction target remains U_r, computed before input masking. Each representation has width 128. The squared cosine reconstruction loss is evaluated over the nodes selected for both masking operations:

$$\mathcal{L}_{\mathrm{rec}} = \sum_{r\in\{c,m\}} \frac{1}{\max(1, |\Omega_r|)} \sum_{i\in\Omega_r}\left(1 - \cos\left(\widehat{U}_r[i,:], U_r[i,:]\right)\right)^2, \tag{26}$$

where cos denotes cosine similarity with numerical stabilization for small norms. The target U_r is recomputed by the trainable backbone at every iteration. Gradients propagate through both the reconstructed representations and the target representations. This loss therefore provides an auxiliary reconstruction constraint in the learned representation space. If a sampled set is empty, its reconstruction-loss contribution is zero and supervised training proceeds. After computing the per-node reconstruction errors $\ell_{r,i}^{(t)}$, update the sampling state:

$$d_{r,i}^{(t)} = \begin{cases} \mu d_{r,i}^{(t-1)} + (1-\mu)\,\mathrm{sg}(\ell_{r,i}^{(t)}), & i \in \Omega_r^{(t)}, \\ d_{r,i}^{(t-1)}, & i \notin \Omega_r^{(t)}, \end{cases} \tag{27}$$

where $\ell_{r,i}^{(t)} = (1 - \cos(\widehat{U}_r[i,:], U_r[i,:]))^2$, sg denotes stop-gradient, and the proposed error-state momentum is $\mu = 0.9$. Only masked-node errors update the state, which is used from the next iteration. The sampler has no trainable parameters and receives no gradients. Detaching this state leaves the target-side gradients of the reconstruction objective intact. The loss averages errors over the adaptively selected nodes, without inverse-probability correction. Input masking and re-mask decoding share the same sampled set.

For supervised association learning, a type-specific three-layer perceptron maps U_r to $Z_r \in \mathbb{R}^{n_r \times 64}$. Its successive output widths are 256, 128, and 64, with a ReLU activation after each layer. For a training pair (c_i, m_j), the pair feature and association logit are

$$\phi_{ij} = [Z_c[i,:] \| Z_m[j,:]] \in \mathbb{R}^{128}, \qquad s_{ij} = \mathbf{w}^\top \phi_{ij} + b. \tag{28}$$

Let $\mathcal{T}_f$ contain the labeled training pairs in fold f, with labels $y_{ij} \in \{0, 1\}$. The representation model minimizes

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + 0.1\mathcal{L}_{\mathrm{rec}}, \qquad \mathcal{L}_{\mathrm{BCE}} = \frac{1}{|\mathcal{T}_f|} \sum_{(i,j)\in\mathcal{T}_f} \mathrm{BCEWithLogits}(s_{ij}, y_{ij}). \tag{29}$$

After representation learning, the trained neural model is held fixed and node embeddings are extracted in evaluation mode. A standardizer is fitted on the training pair features and applied to the pair representations supplied to CatBoost. CatBoost is trained independently on these fixed features. The linear association head is used during representation learning, while CatBoost provides the final interaction scores. The classification path uses U_r before masking; the reconstruction output enters the auxiliary loss.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate SMASH on three benchmark datasets: CMI-9905, CMI-9589, and CMI-20208, curated from the circBank and circR2Cancer databases (Liu et al., 2019; Lan et al., 2020). The CMI-9905 dataset contains 9,905 interaction pairs involving 2,346 circRNAs and 962 miRNAs, while CMI-9589 includes 9,589 interaction pairs with 2,115 circRNAs and 821 miRNAs. The largest dataset, CMI-20208, consists of 20,208 interaction pairs spanning 3,569 circRNAs and 1,152 miRNAs.

Known circRNA–miRNA interactions were treated as positive samples. Candidate negatives were selected from unobserved pairs using IntaRNA screening, with one candidate negative retained for each positive sample to form a 1:1 positive-to-negative ratio. The resulting pairs were partitioned using association-level five-fold cross-validation. For each fold, held-out positive associations were removed before reconstructing $Y^{(f)}$, $\mathcal{H}^{(f)}$, and $G_r^{\mathrm{init},(f)}$. Node inputs combined training association profiles with fused sequence, structural, and composition similarities. The encoder, feature standardizer, and CatBoost classifier were fitted using the training fold, and held-out pairs were used only for evaluation. Appendix C describes the negative-sample construction and random-negative control. Seven standard metrics were adopted, including area under the receiver operating characteristic curve (AUC), area under the precision-recall curve (AUPR), accuracy, sensitivity, precision, specificity, and Matthews correlation coefficient (MCC). Practical runtime and peak memory measurements are reported in Table 7.

4.2 Performance Evaluation

Table 1 summarizes the mean and standard deviation of each metric across the five folds. Appendix F provides the individual fold results.

On CMI-9905, SMASH achieved a mean AUC of 0.9470 and AUPR of 0.9487. CMI-9589 yielded the highest mean AUC and AUPR among the three datasets, at 0.9604 and 0.9596, respectively. On CMI-20208, the corresponding values were 0.9572 and 0.9566. Mean accuracy ranged from 88.47% to 90.55% across the three datasets.

Table 1: The performance achieved by SMASH on the CMI-9905, CMI-9589 and CMI-20208 datasets.

Datasets	Statistic	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
CMI-9905	Mean	0.9470	0.9487	88.47	88.47	88.48	89.20	76.93
	SD	0.0032	0.0025	0.58	0.76	0.92	0.77	1.16
CMI-9589	Mean	0.9604	0.9596	90.55	92.03	88.90	90.21	81.04
	SD	0.0037	0.0052	0.67	0.98	0.50	0.46	1.36
CMI-20208	Mean	0.9572	0.9566	90.10	91.52	88.51	89.86	80.14
	SD	0.0052	0.0067	0.39	0.45	0.90	0.70	0.79

4.3 Ablation Experiments

For w/o Channel Attention, we replace parameter-free channel attention with equal-weight averaging. For w/o Multi-scale HGCN, we remove the multi-scale hypergraph convolution module. For w/o GraphMAE, we remove the GraphMAE reconstruction branch while retaining the remaining components.

Figure 2 presents the ablation results. The complete SMASH model obtains higher scores than all three ablated variants. Removing GraphMAE produces the largest performance decline. Removing channel attention decreases AUC by 0.0429 and AUPR by 0.0418. Removing Multi-scale HGCN decreases AUC by 0.0316 and AUPR by 0.0316. The latter result supports the contribution of the multi-scale representation module. The influence of propagation depth is examined further in Appendix H.

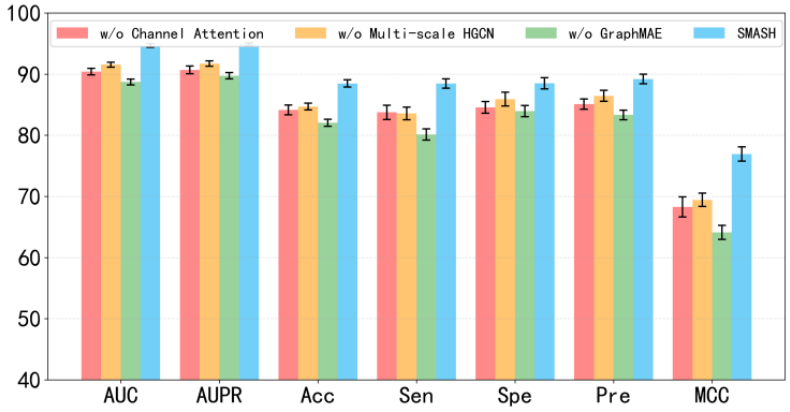

Figure 2: Performance comparison of ablation experiments.

4.4 Comparison with Baseline Methods

Table 2 compares SMASH with published results for BioDGW-CMI (Wang et al., 2023), BEROLECMI (Wang et al., 2024), and MRHRL (Yin et al., 2024) on CMI-9905. The AUC and AUPR values obtained by SMASH were numerically higher than those reported for MRHRL by

0.0194 and 0.0091, respectively. Baseline results were taken from the original publications, each reporting five-fold cross-validation under its respective evaluation protocol.

Table 2: Comparison with published baseline results on CMI-9905.

Methods	BioDGW-CMI	BEROLECMI	MRHRL	SMASH
AUC	0.9026	0.9104	0.9276	**0.9470**
AUPR	0.8962	0.9086	0.9396	**0.9487**

4.5 Case Study

We conduct a literature-based case analysis of the 20 highest-scoring CMI pairs on CMI-9905. All candidate pairs were excluded from the supervised training set and from the fold-specific interaction matrix before representation learning and ranking. Among the top 20 predicted CMI pairs ranked by prediction score, PubMed literature search confirms experimental validation for 19 pairs. Table 3 reports the validation status of all 20 pairs. For instance, Song et al. (2022) validated that *hsa_circ_0001666* sponges *hsa-miR-661* to restore tumor suppressor PPM1L expression, suppressing esophageal squamous cell carcinoma progression. Similarly, Zhou et al. (2020) confirmed *hsa_circ_0006916* overexpression in hepatocellular carcinoma, where it sponges *hsa-miR-337-3p* to modulate tumor progression. The association between *hsa_circ_0013876* and *hsa-miR-188-3p* remains a candidate for experimental follow-up.

Table 3: Top 20 predicted circRNA–miRNA pairs on CMI-9905 and their reported validation status.

circRNA	miRNA	Evid.	circRNA	miRNA	Evid.
hsa_circ_0081065	hsa-miR-665	Verified	hsa_circ_0058092	hsa-miR-217	Verified
hsa_circ_0001666	hsa-miR-661	Verified	hsa_circ_0043278	hsa-miR-455-3p	Verified
hsa_circ_0011385	hsa-miR-149-5p	Verified	hsa_circ_0025202	hsa-miR-197-3p	Verified
hsa_circ_0003258	hsa-miR-653-5p	Verified	hsa_circ_0013876	hsa-miR-188-3p	Unverified
hsa_circ_0000285	hsa-miR-127-5p	Verified	hsa_circ_0014130	hsa-miR-545-3p	Verified
hsa_circ_0081069	hsa-miR-665	Verified	hsa_circ_0000977	hsa-miR-874-3p	Verified
hsa_circ_0006916	hsa-miR-337-3p	Verified	hsa_circ_0005100	hsa-miR-145-5p	Verified
hsa_circ_0005227	hsa-miR-27a-3p	Verified	hsa_circ_0000073	hsa-miR-1184	Verified
hsa_circ_0001666	hsa-miR-330-5p	Verified	hsa_circ_0001946	hsa-miR-7-5p	Verified
hsa_circ_0014130	hsa-miR-665	Verified	hsa_circ_0078767	hsa-miR-188-3p	Verified

5 Conclusion

SMASH learns circRNA-miRNA interaction representations from association profiles and fused RNA similarities. Adaptive propagation is constructed from complete within-type similarity profiles. The multi-scale hypergraph backbone retains depth-specific representations and learns node-specific aggregation weights. A reconstruction-error-guided sampler selects backbone representations for asking, and an auxiliary branch reconstructs them through re-mask decoding. Representation learning jointly optimizes masked reconstruction and supervised association prediction, followed by CatBoost classification using the learned embeddings. SMASH achieved AUC/AUPR values of 0.9470/0.9487, 0.9604/0.9596, and 0.9572/0.9566 on CMI-9905, CMI-9589, and CMI-20208, respectively. Component ablations examine the contribution of each module. RNA-disjoint evaluations further characterize performance for RNAs excluded from supervised training pairs, with their features remaining available during representation learning.

Several limitations remain. SMASH currently focuses solely on CMI prediction. Extending its framework to capture broader molecular interaction networks remains an important direction for future research. Future work will investigate more scalable implementations of adaptive propagation and evaluate the model on additional molecular interaction tasks.

AI USE STATEMENT

Generative AI tools were used only for editorial assistance during manuscript preparation, mainly for proofreading and improving textual clarity. The research methodology and experimental work were carried out by the authors without AI assistance. All reported results and scientific conclusions were independently verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Rommie E Amaro and Adrian J Mulholland. Multiscale methods in drug design bridge chemical and biological complexity in the search for cures. *Nature Reviews Chemistry*, 2(4):0148, 2018.

Wele Gedara Chaminda Bandara, Naman Patel, Ali Gholami, Mehdi Nikkhah, Motilal Agrawal, and Vishal M. Patel. Adamae: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14507–14517, June 2023.

Guorong Chen, Hongying Qi, Li Jiang, Shijie Sun, Junhai Zhang, Jiali Yu, Fang Liu, Yanli Zhang, and Shiyu Du. Integrating single-cell rna-seq and machine learning to dissect tryptophan metabolism in ulcerative colitis. *Journal of translational medicine*, 22(1):1121, 2024.

Jianlin Cheng, Allison N Tegge, and Pierre Baldi. Machine learning methods for protein structure prediction. *IEEE reviews in biomedical engineering*, 1:41–49, 2008.

Zhen Cui, Yan Wu, Qin-Hu Zhang, Si-Guo Wang, and Zhen-Hao Guo. Npenn: a noise perturbation ensemble neural network for microbiome disease phenotype prediction. *IEEE Journal of Biomedical and Health Informatics*, 29(3):2210–2221, 2025. doi: 10.1109/JBHI.2024.3507789.

Haifeng Dong, Jianping Lei, Lin Ding, Yongqiang Wen, Huangxian Ju, and Xueji Zhang. Microrna: function, detection, and bioanalysis. *Chemical reviews*, 113(8):6207–6233, 2013.

Yanhao Fan, Che Zhang, Xiaowen Hu, Zhijian Huang, and Lei Deng. Agclnda: enhancing the prediction of ncrna-drug resistance association using adaptive graph contrastive learning. *IEEE Journal of Biomedical and Health Informatics*, 29(6):4524–4533, 2025.

Lu-Xiang Guo, Zhu-Hong You, Lei Wang, Chang-Qing Yu, Bo-Wei Zhao, Zhong-Hao Ren, and Jie Pan. A novel circrna-mirna association prediction model based on structural deep neural network embedding. *Briefings in Bioinformatics*, 23(5):bbac391, 2022.

Lu-Xiang Guo, Lei Wang, Zhu-Hong You, Chang-Qing Yu, Meng-Lei Hu, Bo-Wei Zhao, and Yang Li. Likelihood-based feature representation learning combined with neighborhood information for predicting circrna–mirna associations. *Briefings in Bioinformatics*, 25(2):bbae020, 2024.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 594–604, 2022.

Chen Jiang, Lei Wang, Changqing Yu, Zhuhong You, Xinfei Wang, Mengmeng Wei, and Mianshuo Lu. MuGNet-CMI: Multi-head hybrid graph neural network for predicting circRNA-miRNA interactions with global high-order and local low-order information. *IEEE Transactions on Big Data*, 12(1):159–173, 2026. doi: 10.1109/TBDATA.2025.3604175.

Mingming Jin, Tailei Yuan, Kaisai Tian, Jingjing Li, Qingqing Huang, Yongbin Chi, and Gang Huang. Oncogenic circ-slc16a1 promotes progression of non-small cell lung cancer via regulation of the mir-1287-5p/profilin 2 axis. *Cellular & Molecular Biology Letters*, 29(1):43, 2024.

Anusuya Krishnan and Isaias Mehari Ghebrehiwet. Oppi-grf: Optimizing protein-protein interaction prediction with graph-based representation and fusion. In *Latinx in AI@ NeurIPS 2024*, 2024.

Wei Lan, Mingrui Zhu, Qingfeng Chen, Baoshan Chen, Jin Liu, Min Li, and Yi-Ping Phoebe Chen. Circr2cancer: a manually curated database of associations between circrnas and cancers. *Database*, 2020:baaa085, 2020.

Haoyang Li, Xin Wang, Zeyang Zhang, Zongyuan Wu, Linxin Xiao, and Wenwu Zhu. Self-supervised masked graph autoencoder via structure-aware curriculum. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36215–36235. PMLR, 2025.

Julong Li, Jianrui Chen, Zhihui Wang, and Xiujuan Lei. Horda: Learning higher-order structure information for predicting rna–disease associations. *Artificial Intelligence in Medicine*, 148:102775, 2024.

Rui Li, Xin Yuan, Mohsen Radfar, Peter Marendy, Wei Ni, Terrence J O'Brien, and Pablo M Casillas-Espinosa. Graph signal processing, graph neural network and graph learning on biological data: a systematic review. *IEEE Reviews in Biomedical Engineering*, 16:109–135, 2023. doi: 10.1109/RBME.2021.3122522.

Yao Liang, Siwei Xie, Xuqi Zheng, Xinyu Wu, Sijin Du, and Yutong Jiang. Predicting higher risk factors for covid-19 short-term reinfection in patients with rheumatic diseases: a modeling study based on xgboost algorithm. *Journal of Translational Medicine*, 22(1):1144, 2024.

Lu Lin, Wen Xue, Xindian Wei, Wenjun Shen, Cheng Liu, Si Wu, and Hau-San Wong. Sctrans: Multi-scale scrna-seq sub-vector completion transformer for gene-selective cell type annotation. In *IJCAI*, pp. 5954–5962, 2024.

Ming Liu, Qian Wang, Jian Shen, Burton B Yang, and Xiangming Ding. Circbank: a comprehensive database for circrna with standard nomenclature. *RNA biology*, 16(7):899–905, 2019.

Weiwei Liu, Lu Zheng, Rongguiyi Zhang, Ping Hou, Jiakun Wang, Linquan Wu, and Jing Li. Circ-zeb1 promotes pik3ca expression by silencing mir-199a-3p and affects the proliferation and apoptosis of hepatocellular carcinoma. *Molecular cancer*, 21(1):72, 2022.

Yue Luo and Lei Deng. Dpmgcda: Deciphering circrna–drug sensitivity associations with dual perspective learning and path-masked graph autoencoder. *Journal of Chemical Information and Modeling*, 64(10):4359–4372, 2024.

Yu Ma, Ting Wang, Xudong Zhang, Pinghan Wang, and Fangyi Long. The role of circular rnas in regulating resistance to cancer immunotherapy: mechanisms and implications. *Cell Death & Disease*, 15(5):312, 2024.

Giuseppina Pisignano, David C Michael, Tanvi H Visal, Radu Pirlog, Michael Ladomery, and George A Calin. Going circular: history, present, and future of circrnas in cancer. *Oncogene*, 42 (38):2783–2800, 2023.

Omer Rochman Sharabi and Gilles Louppe. Trick or treat? evaluating stability strategies in graph network-based simulators. In *Machine Learning and the Physical Sciences, NeurIPS 2023*, 2023.

Huan Song, Dan Tian, Jian Sun, Xuhua Mao, Weimin Kong, Dian Xu, Ye Ji, Beibei Qiu, Mengyao Zhan, and Jianming Wang. circfam120b functions as a tumor suppressor in esophageal squamous cell carcinoma via the mir-661/ppm1l axis and the pkr/p38 mapk/emt pathway. *Cell Death & Disease*, 13(4):361, 2022.

Xin-Fei Wang, Chang-Qing Yu, Zhu-Hong You, Yan Qiao, Zheng-Wei Li, and Wen-Zhun Huang. An efficient circrna-mirna interaction prediction model by combining biological text mining and wavelet diffusion-based sparse network structure embedding. *Computers in Biology and Medicine*, 165:107421, 2023.

Xin-Fei Wang, Chang-Qing Yu, Zhu-Hong You, Yan Wang, Lan Huang, Yan Qiao, Lei Wang, and Zheng-Wei Li. Berolecmi: a novel prediction method to infer circrna-mirna interaction from the role definition of molecular attributes and biological networks. *BMC bioinformatics*, 25(1):264, 2024.

Yuzhe Wang, Yibo Li, Jiaxiao Chen, and Luhua Lai. Modeling protein–ligand interactions for drug discovery in the era of deep learning. *Chemical Society Reviews*, 54(23):11141–11183, 2025. doi: 10.1039/D5CS00415B.

Meng-Meng Wei, Lei Wang, Bo-Wei Zhao, Xiao-Rui Su, Zhu-Hong You, and De-Shuang Huang. Integrating transformer and graph attention network for circrna-mirna interaction prediction. *IEEE Journal of Biomedical and Health Informatics*, 29(8):6105–6113, 2025. doi: 10.1109/JBHI.2025.3561197.

Ping Xuan, Siyuan Lu, Hui Cui, Shuai Wang, Toshiya Nakaguchi, and Tiangang Zhang. Learning association characteristics by dynamic hypergraph and gated convolution enhanced pairwise attributes for prediction of disease-related lncrnas. *Journal of Chemical Information and Modeling*, 64(8):3569–3578, 2024.

Wenjing Yin, Shudong Wang, Yuanyuan Zhang, Sibo Qiao, Wenhao Wu, and Hengxiao Li. Multirelational hypergraph representation learning for predicting circrna-mirna associations. *Journal of Chemical Information and Modeling*, 64(21):8349–8360, 2024.

Yalin Yu, Enneng Yang, Guibing Guo, Linying Jiang, and Xingwei Wang. Basket representation learning by hypergraph convolution on repeated items for next-basket recommendation. In *IJCAI*, pp. 2415–2422, 2023.

Lukáš Zahradník, Jan Neumann, and Gustav Šír. A deep learning blueprint for relational databases. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

Fan Zhang, Jiajia Jiang, Hui Qian, Yongmin Yan, and Wenrong Xu. Exosomal circrna: emerging insights into cancer progression and clinical application potential. *Journal of hematology & oncology*, 16(1):67, 2023.

Jingwei Zhang and Srinivasan Sivanandan. Channelsformer: A channel agnostic vision transformer for multi-channel cell painting images. In *NeurIPS 2025 Workshop for Imageomics: Discovering Biological Knowledge from Images Using AI*, 2025.

Liwen Zhang, Xinxin Liu, Yu Zhang, Lang Qin, Shijia Pan, Xueqi Yan, Sen Dong, Zerong Feng, Song-jia Fan, Rusong Zhao, et al. Peripheral blood rna modifications as a novel diagnostic signature for polycystic ovary syndrome. *Science China Life Sciences*, 68(11):3231–3241, 2025.

Xinyu Zhang, Mingzhe Liu, Zhen Li, Linlin Zhuo, Xiangzheng Fu, and Quan Zou. Fusion of multi-source relationships and topology to infer lncrna-protein interactions. *Molecular Therapy Nucleic Acids*, 35(2):102187, 2024a. doi: 10.1016/j.omtn.2024.102187.

Xinyu Zhang, Yifei Wang, Qinqin Wei, Shiyue He, Adil Salhi, and Bin Yu. Drbppred-gat: Accurate prediction of dna-binding proteins and rna-binding proteins based on graph multi-head attention network. *Knowledge-Based Systems*, 285:111354, 2024b.

Xiao-Yong Zhou, Hui Yang, Yang-Qiu Bai, Xiu-Ling Li, Shuang-Yin Han, and Bing-Xi Zhou. hsa_circ_0006916 promotes hepatocellular carcinoma progression by activating the mir-337-3p/stat3 axis. *Cellular & Molecular Biology Letters*, 25(1):47, 2020.

A Additional Experimental Details

This appendix provides supplementary experimental details for SMASH, including the experimental protocol, training configuration, evaluation metrics, computational complexity, and reproducibility settings. The methodological components of SMASH, including multi-view similarity integration, hypergraph-based representation learning, self-supervised reconstruction, and association prediction, are formally defined in the main text. To avoid redundant notation, this appendix focuses on the practical experimental settings used to obtain the reported results.

SMASH follows a unified learning pipeline for circRNA–miRNA association prediction. For each RNA type, the sequence, structural, and composition similarity matrices are fused before hypergraph propagation. The training-fold association matrix is then used to construct the initial joint hypergraph. The multi-scale hypergraph backbone processes these inputs to obtain circRNA and miRNA node embeddings. The corresponding embeddings are concatenated to construct pairwise representations for association prediction. In addition, a masked reconstruction objective is optimized jointly with the supervised association loss during representation learning.

For multi-view similarity integration, three complementary similarity sources are considered for each RNA type. The three views encode sequence, predicted structural, and dinucleotide-composition similarity and are fused before hypergraph learning. The fused similarity matrix is computed from channel weights derived from the three similarity matrices. The same integration strategy is applied independently to circRNAs and miRNAs so that both molecular types are represented in a consistent feature space.

The relational structure between circRNAs and miRNAs is represented using a hypergraph formulation. Each RNA-centered hyperedge groups an RNA with its observed interaction partners, enabling propagation among nodes that share a hyperedge. This is important for biological association prediction, where one circRNA may be related to multiple miRNAs and one miRNA may also participate in interactions with multiple circRNAs. The hypergraph representation therefore provides a natural way to encode group-level relational patterns beyond isolated pairwise links.

To capture structural information at different propagation depths, SMASH adopts a multi-scale hypergraph representation module. Each scale characterizes node information from a different neighborhood range, and the scale-specific representations are adaptively combined to form the final node embeddings. This multi-scale design helps the model balance local association patterns and broader structural context, which is particularly useful for sparse circRNA–miRNA association matrices. Dropout and nonlinear transformations are used during representation learning to reduce overfitting and improve generalization across folds.

An auxiliary masked reconstruction objective is incorporated during representation learning. The adaptive sampler uses historical reconstruction errors to determine node selection probabilities after a uniform-sampling warm-up. Selected backbone representations are replaced by a learnable token before auxiliary encoding. The corresponding encoder outputs are then set to zero before hypergraph decoding. The decoder reconstructs the backbone representations computed before masking. The squared cosine reconstruction loss is evaluated at the selected nodes and combined with the supervised association prediction loss to train the representation model.

For each candidate circRNA-miRNA pair, the corresponding 64-dimensional circRNA and miRNA embeddings are concatenated to form a 128-dimensional pairwise representation. The pairwise features are standardized using statistics fitted on the training data and are subsequently used to train the CatBoost classifier for CMI prediction.

A.1 Adaptive Propagation Operator

For each RNA type r, the adaptive propagation module receives the complete fused within-type similarity matrix,

$$R_c = \widetilde{S}_c \in \mathbb{R}^{n_c \times n_c}, \qquad R_m = \widetilde{S}_m \in \mathbb{R}^{n_m \times n_m}. \tag{A.1}$$

Each row contains the similarities between one RNA and all RNAs of the same type, following the fixed node order used by the backbone. Thus, R_r has n_r input features per node. The backbone retains $X_c = [Y^{(f)} \| \widetilde{S}_c]$ and $X_m = [(Y^{(f)})^\top \| \widetilde{S}_m]$. The initial propagation operator is constructed

from the training association matrix, while the adaptive operator is constructed from transformed similarity profiles.

The feature transformation is

$$E_r = \mathrm{ReLU}(R_r W_{r,1} + \mathbf{1} b_{r,1}^\top) W_{r,2} + \mathbf{1} b_{r,2}^\top, \tag{A.2}$$

where $W_{r,1} \in \mathbb{R}^{n_r \times 128}$, $W_{r,2} \in \mathbb{R}^{128 \times 128}$, and both biases have 128 entries. Each RNA type has its own parameters. The second linear layer produces E_r directly. Pairwise scores are the unnormalized inner products

$$S_r^{\mathrm{adp}} = E_r E_r^\top. \tag{A.3}$$

For each row, we select the $k+1$ entries with the largest scores, including the diagonal among the eligible entries, with $k = 10$. Writing the returned indices in descending score order gives

$$(j_{r,i,1}, \ldots, j_{r,i,k+1}) = \mathrm{TopKIndices}_{k+1}(S_r^{\mathrm{adp}}[i,:]), \qquad \mathcal{J}_{r,i} = \{j_{r,i,2}, \ldots, j_{r,i,k+1}\}. \tag{A.4}$$

Tied indices follow the order returned by the top-k implementation. The largest returned entry is discarded. This rule does not guarantee that the discarded entry is the diagonal, since an off-diagonal inner product can exceed a self-inner-product. The incidence entries are

$$B_r[i,j] = \mathbb{I}\{i = j \text{ or } j \in \mathcal{J}_{r,i}\}. \tag{A.5}$$

Each row has k or $k+1$ nonzero entries, depending on whether its diagonal index is already retained. Rows index vertices and columns index hyperedges in the subsequent normalization. Consequently, column j contains node j and the nodes whose retained index sets contain j. This row-wise construction determines the orientation of the resulting incidence matrix.

A separate network produces the incidence-row coefficients:

$$\boldsymbol{\xi}_r = \mathrm{sigmoid}\big(\mathrm{ReLU}(E_r W_{r,a} + \mathbf{1} b_{r,a}^\top) w_{r,b} + b_{r,b} \mathbf{1}\big), \qquad \widehat{B}_r = \mathrm{diag}(\boldsymbol{\xi}_r) B_r, \tag{A.6}$$

where $W_{r,a} \in \mathbb{R}^{128 \times 32}$, $b_{r,a} \in \mathbb{R}^{32}$, $w_{r,b} \in \mathbb{R}^{32}$, and $b_{r,b}$ is a scalar. Broadcasting the coefficients along each row implements this multiplication. The coefficients weight vertex incidences within every column containing that vertex.

Define the weighted degrees and inverse-degree factors by

$$\begin{aligned} d_{v,r}(i) &= \sum_j \widehat{B}_r[i,j], & d_{e,r}(j) &= \sum_i \widehat{B}_r[i,j], \\ Q_{v,r} &= \mathrm{diag}(q_{1/2}(d_{v,r}(i))), & Q_{e,r} &= \mathrm{diag}(q_1(d_{e,r}(j))). \end{aligned} \tag{A.7}$$

The function q_α follows the zero-degree convention in the main text. The complete adaptive operator is

$$\mathcal{A}_r(R_r) = G_r^{\mathrm{adaptive}} = Q_{v,r} \widehat{B}_r Q_{e,r} \widehat{B}_r^\top Q_{v,r}. \tag{A.8}$$

Self-membership is already represented in B_r. The final operator uses the expression in Eq. (A.8) directly, with unit external hyperedge weights. Its support can be denser than that of B_r, and the implementation stores and multiplies these matrices as dense tensors.

The module is recomputed at each forward pass and shared across the propagation layers and auxiliary reconstruction branch within that pass. During representation learning, gradients reach the feature transformation and coefficient network through $\boldsymbol{\xi}_r$, the weighted degrees, and the normalized operator. The hard selection indices receive no gradient; the selected similarity values are discarded when constructing the binary matrix. Parameter updates can change the rankings in later forward passes. During evaluation, the same operator is computed using fixed learned parameters and the training-fold inputs. Adaptive propagation and the proposed reconstruction-error-based mask sampler use separate computations.

B Algorithmic Description

Algorithm 1 describes representation learning and classification for a fold. The training CMI matrix $Y^{(f)}$ and fused similarity matrices define the backbone inputs. The initial operators are obtained from Eqs. (6)–(9). $\mathcal{A}_r$ is defined in Appendix A.1. Its inputs are $R_c = \widetilde{S}_c$ and $R_m = \widetilde{S}_m$, containing

complete within-type similarity profiles in the backbone node order. $\boldsymbol{\xi}_r$ denotes incidence-row coefficients and is separate from scale attention a_r. U_r is the 128-dimensional backbone output, H_r^{enc} is the auxiliary encoder output, and Z_r is the 64-dimensional embedding used for classification. $\widetilde{H}_r^{\mathrm{enc}}$ denotes the encoder output after zeroing the rows indexed by Ω_r for re-mask decoding. The joint incidence matrix $\mathcal{H}^{(f)}$ and auxiliary encoding H_r^{enc} have separate roles. The functions σ, Drop, and a_r denote LeakyReLU with slope 0.25, dropout with rate 0.2, and the shared scale-attention MLP. Θ collects the backbone, adaptive modules, node projection heads, auxiliary encoders and decoders, mask tokens, and linear classification head. Error states are separate from Θ and are reset to one in every fold. The sampler uses $T_0 = 5$, $\eta = 0.5$, $\mu = 0.9$, and $\epsilon = 10^{-8}$.

Algorithm 1 Representation Learning and Classification in SMASH for Fold f

Input: Training matrix $Y^{(f)}$, fused similarities $\widetilde{S}_c, \widetilde{S}_m$, labeled training pairs $\mathcal{T}_f$
Output: Representation model f_Θ, standardizer q, CatBoost g_Ψ
Initialize Θ and $d_{r,i}^{(0)} = 1$; set $L = 3$, $\rho = 0.5$, and $T = 30$
Construct $\mathcal{H}^{(f)}$ using Eq. (6)
Compute $G_c^{\mathrm{init}}, G_m^{\mathrm{init}}$ using Eq. (9)
$X_c \leftarrow [Y^{(f)} \| \widetilde{S}_c]$; $X_m \leftarrow [(Y^{(f)})^\top \| \widetilde{S}_m]$
for $t = 1$ to T **do**
  for each RNA type $r \in \{c, m\}$ **do**
    $R_r \leftarrow \widetilde{S}_r$; compute E_r, B_r using Eqs. (A.2)–(A.5)
    Compute $\boldsymbol{\xi}_r$ and G_r^{adaptive} using Eqs. (A.6)–(A.8)
    $\bar{G}_r \leftarrow 0.7G_r^{\mathrm{init}} + 0.3G_r^{\mathrm{adaptive}}$; $X_r^0 \leftarrow X_r$
    for $\ell = 1$ to L **do**
      $X_r^\ell \leftarrow \mathrm{Drop}\big(\sigma\big(\bar{G}_r(X_r^{\ell-1}W_r^\ell + \mathbf{1}(b_r^\ell)^\top)\big)\big)$
    end for
    $\beta_{r,i,\ell} \leftarrow \mathrm{Softmax}_\ell(a_r(X_r^\ell[i,:]))$ for each node i
    $F_r[i,:] \leftarrow \sum_{\ell=1}^{L} \beta_{r,i,\ell} X_r^\ell[i,:]$ for each node i
    $U_r \leftarrow \sigma(P_r^{\mathrm{out}}(F_r + P_r^{\mathrm{res}}(X_r)))$
    $Z_r \leftarrow \mathrm{MLP}_r(U_r) \in \mathbb{R}^{n_r \times 64}$
    Compute $p_{r,i}^{(t)}$ using Eq. (20); sample $m_{r,i} \sim \mathrm{Bernoulli}(p_{r,i}^{(t)})$; $\Omega_r \leftarrow \{i : m_{r,i} = 1\}$
    $\widetilde{U}_r \leftarrow \mathrm{Clone}(U_r)$; replace rows in Ω_r by $\mathbf{t}_r$
    $H_r^{\mathrm{enc}} \leftarrow \bar{G}_r(\widetilde{U}_r W_r^{\mathrm{enc}} + \mathbf{1}(b_r^{\mathrm{enc}})^\top)$
    $\widetilde{H}_r^{\mathrm{enc}} \leftarrow \mathrm{Clone}(H_r^{\mathrm{enc}})$
    $\widetilde{H}_r^{\mathrm{enc}}[i,:] \leftarrow \mathbf{0}$ for all $i \in \Omega_r$
    $\widehat{U}_r \leftarrow \bar{G}_r(\widetilde{H}_r^{\mathrm{enc}} W_r^{\mathrm{dec}} + \mathbf{1}(b_r^{\mathrm{dec}})^\top)$
    $\mathcal{L}_r \leftarrow \max(1, |\Omega_r|)^{-1} \sum_{i \in \Omega_r} (1 - \cos(\widehat{U}_r[i,:], U_r[i,:]))^2$
    Retain target-side gradients through U_r in $\mathcal{L}_r$
    Update detached error states using Eq. (27)
  end for
  $\phi_{ij} \leftarrow [Z_c[i,:] \| Z_m[j,:]]$ for $(i,j) \in \mathcal{T}_f$
  $s_{ij} \leftarrow \mathbf{w}^\top \phi_{ij} + b$; $\mathcal{L}_{\mathrm{BCE}} \leftarrow \mathrm{mean}_{\mathcal{T}_f} \mathrm{BCEWithLogits}(s_{ij}, y_{ij})$
  Update Θ with Adam using $\mathcal{L}_{\mathrm{BCE}} + 0.1(\mathcal{L}_c + \mathcal{L}_m)$
end for
Extract Z_c, Z_m in evaluation mode from the backbone and node projection heads
$\phi_{ij} \leftarrow [Z_c[i,:] \| Z_m[j,:]] \in \mathbb{R}^{128}$ for each labeled pair
Fit standardizer q on training pair features
Split standardized training pairs into 85% fitting and 15% validation subsets
Fit CatBoost g_Ψ on the fitting subset with validation-based early stopping
return f_Θ, q, g_Ψ

C EXPERIMENTAL SETTINGS

The primary within-dataset evaluation and random-negative control followed an association-level five-fold cross-validation protocol. Positive and candidate-negative pairs were partitioned jointly into five folds. Four folds were used for training and the remaining fold was used for evaluation. For each fold, held-out positive associations were excluded before the interaction matrix and hypergraphs were reconstructed. The encoder, feature standardizer, and CatBoost classifier were fitted using the training fold only. This procedure was repeated until every fold had been used for evaluation, and the final performance was reported as the mean across the five folds.

All experiments were run on a workstation equipped with an Intel Core i5-12600KF CPU, 32 GB of system memory, and an NVIDIA GeForce RTX 4060 Ti GPU.

Negative-sample construction. Known circRNA–miRNA interactions were used as positive samples. Unobserved pairs were screened using IntaRNA because the absence of an annotated interaction does not establish a biological non-interaction. For each unobserved pair, the mature miRNA was supplied as the query and the circRNA as the target. IntaRNA evaluated sequence complementarity, seed pairing, duplex stability, and target-site accessibility. Pairs with weak predicted binding evidence were retained as candidate negatives. An equal number of positive and candidate-negative pairs was used in each dataset. The same ratio and fold-specific reconstruction protocol were used in all SMASH experiments reported in this paper.

During representation learning, we trained the model for 30 epochs using Adam with a learning rate of 1×10^{-4}. The dropout rate is set to 0.2. The number of hypergraph scales is set to 3, and the final embedding dimension for each RNA type is set to 64. Therefore, each circRNA–miRNA pair is represented by a 128-dimensional pairwise feature after combining the two node embeddings.

For the uniform-masking configuration, each node is selected independently with probability $\rho = 0.5$. The proposed adaptive sampler preserves this mean probability and applies Eqs. (20) and (27). Input masking replaces selected backbone representations with a learnable token. Re-mask decoding replaces the corresponding encoder outputs with zero vectors, using the same node set. The squared cosine reconstruction loss has weight 0.1.

After representation learning, the learned pairwise features are standardized and used to train the final classifier. CatBoost is configured with 1500 boosting iterations, learning rate 0.04, tree depth 6, and L_2 leaf regularization coefficient 6.0. Logloss is used as the training objective, and AUC is used as the validation metric. Early stopping is applied with 50 rounds.

The main hyperparameters used in the experiments are summarized in Table 4.

Table 4: Key model hyperparameters of SMASH.

Hyperparameter	Value
Number of hypergraph scales	3
Adaptive propagation feature dimension	128
Adaptive propagation rank parameter k	10
Initial/adaptive propagation coefficients	$0.7/0.3$
Final embedding dimension per RNA type	64
Mean node masking probability ρ	0.5
Masking warm-up T_0	5 iterations
Adaptive masking strength η	0.5
Error-state momentum μ	0.9
Reconstruction loss weight	0.1

Random-negative control. We conducted a control experiment on CMI-9905 using randomly sampled unobserved pairs as candidate negatives. The experiment used the same 1:1 positive-to-negative ratio, five-fold partition, model configuration, and training procedure as the primary experiment. Table 5 reports the fold-level results, including F1. The random-negative control

achieved mean AUC and AUPR values of 0.9363 and 0.9354, respectively. Mean accuracy was 86.99%, and mean F1 was 87.32%.

Table 5: Fold-level performance on CMI-9905 using randomly sampled candidate negatives. Values in the last row are reported as mean ± sample standard deviation across five folds. Standard deviations were calculated from the reported fold-level values.

Fold	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)	F1(%)
1	0.9379	0.9398	87.73	87.08	88.44	89.01	75.47	88.03
2	0.9397	0.9381	87.55	86.12	89.09	89.46	75.16	87.76
3	0.9283	0.9274	85.35	85.31	85.40	86.27	70.68	85.79
4	0.9386	0.9385	87.44	86.57	88.38	88.91	74.91	87.72
5	0.9371	0.9334	86.90	86.72	87.08	87.83	73.77	87.27
Mean	0.9363	0.9354	86.99	86.36	87.68	88.30	74.00	87.32
± SD	±0.0046	±0.0051	±0.97	±0.68	±1.47	±1.28	±1.96	±0.89

Table 6 compares the random-negative control with the primary IntaRNA-screened setting. The IntaRNA-screened setting obtained AUC and AUPR values of 0.9470 and 0.9487. These values exceeded the random-negative results by 0.0107 and 0.0133, respectively. The corresponding differences in accuracy and MCC were 1.48 and 2.93 percentage points. Differences were calculated from the reported mean values.

Table 6: Comparison of IntaRNA-screened and randomly sampled candidate negatives on CMI-9905. Differences are calculated from the reported means; differences for percentage-based metrics are expressed in percentage points.

Strategy	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
IntaRNA-screened	0.9470	0.9487	88.47	88.47	88.48	89.20	76.93
Randomly sampled	0.9363	0.9354	86.99	86.36	87.68	88.30	74.00
Screened − random	0.0107	0.0133	1.48	2.11	0.80	0.90	2.93

Random sampling may include unannotated interactions or pairs with plausible binding sites, which introduces ambiguity into the negative class. The performance difference therefore measures the sensitivity of SMASH to negative-sample construction. The primary results should be interpreted together with the IntaRNA screening protocol.

D Evaluation Metrics

We evaluate SMASH using both ranking-based and threshold-based metrics. The ranking-based metrics include the area under the receiver operating characteristic curve (AUC) and the area under the precision–recall curve (AUPR). AUC measures the overall ability of the model to rank positive samples above negative samples. AUPR summarizes the precision–recall trade-off under the candidate-negative protocol. The threshold-based metrics include Accuracy, Recall, Specificity, Precision, F1-score, and Matthews correlation coefficient (MCC).

Let TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. The threshold-based metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{D.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{D.2}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{D.3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{D.4}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{D.5}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{D.6}$$

All threshold-based metrics are computed from binary predictions obtained using a decision threshold of 0.5. AUC and AUPR are computed from the predicted association probabilities.

E Complexity Analysis

The computational cost of SMASH mainly comes from three components: hypergraph construction and refinement, multi-scale hypergraph representation learning, and downstream classification. Let N denote the number of nodes, F denote the hidden feature dimension, and L denote the number of hypergraph scales. For one RNA type with N nodes, the adaptive feature transformation and pairwise scores require $O(N^2F + NF^2)$ operations. The implementation explicitly forms the normalized adaptive operator through dense $N \times N$ matrix multiplications, which incur an $O(N^3)$ cost per forward pass. Rank selection limits the support of the incidence matrix, while the dense implementation retains $O(N^2)$ matrix storage. These costs apply separately to the two RNA types.

For multi-scale hypergraph learning, the propagation operation is repeated across L scales. Under a dense matrix representation, the approximate computational cost is $O(LN^2F)$. In practice, the actual running time depends on the number of circRNAs, the number of miRNAs, the density of the association matrix, and the feature dimension. The self-supervised reconstruction branch has the same order of complexity as the hypergraph representation module because it reuses hypergraph-based encoding and decoding operations.

Practical runtime and memory. We profiled SMASH on the workstation described in Appendix C under the five-fold evaluation protocol. Per-fold time is reported as the mean $\pm$ standard deviation across five folds. Five-fold total denotes the measured wall-clock time for one complete evaluation. GPU memory records peak allocated and reserved memory. Host memory records peak RAM usage.

Table 7: Measured runtime and peak memory usage under five-fold evaluation.

Dataset	Time/fold (s)	Five-fold total (s)	Peak GPU allocated/ reserved (GiB)	Peak host RAM (GiB)
CMI-9905	45.184 $\pm$6.172	226.992	0.406/0.480	3.131
CMI-9589	43.159 $\pm$2.065	216.901	0.344/0.408	2.928
CMI-20208	99.272 $\pm$35.881	497.603	0.825/0.936	4.042

CMI-20208 contains 3,569 circRNAs and required 99.272 ± 35.881 seconds per fold. Its complete five-fold evaluation took 497.603 seconds, corresponding to approximately 8.29 minutes. Peak GPU memory reached 0.825 GiB allocated and 0.936 GiB reserved. Peak host RAM usage was 4.042 GiB. Across the three datasets, the complete five-fold runtime ranged from 216.901 to 497.603 seconds. Peak reserved GPU memory ranged from 0.408 to 0.936 GiB. Peak host RAM ranged from 2.928 to 4.042 GiB.

F Detailed Results for Each Fold

F.1 CMI-9905 Dataset

Table 8 shows the detailed five-fold cross-validation results of SMASH on the CMI-9905 dataset. The model achieved an average AUC of 0.9470 and an average AUPR of 0.9487.

Table 8: Detailed five-fold cross-validation results on CMI-9905.

Fold	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
Fold1	0.9430	0.9455	87.52	88.04	86.97	87.90	75.01
Fold2	0.9445	0.9471	88.36	87.43	89.36	89.83	76.74
Fold3	0.9486	0.9488	88.70	88.54	88.87	89.54	77.38
Fold4	0.9507	0.9519	88.83	89.30	88.33	89.16	77.63
Fold5	0.9483	0.9502	88.96	89.05	88.87	89.59	77.90
Average	**0.9470**	**0.9487**	**88.47**	**88.47**	**88.48**	**89.20**	**76.93**
SD	0.0032	0.0025	0.58	0.76	0.92	0.77	1.16

Figure 3 presents the ROC and PR curves across the five folds on CMI-9905. Fold-level AUC ranged from 0.9430 to 0.9507, while AUPR ranged from 0.9455 to 0.9519. These ranges describe the variation across the evaluated folds.

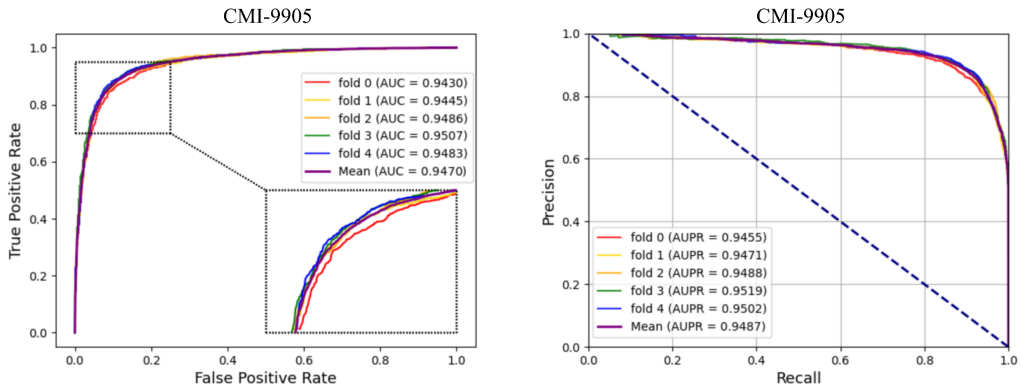

Figure 3: The ROC and PR curves for the CMI-9905 dataset.

F.2 CMI-9589 Dataset

Table 9 lists the five-fold cross-validation results of SMASH on CMI-9589. The model achieved an average AUC of 0.9604 and an average AUPR of 0.9596.

Table 9: Detailed five-fold cross-validation results on CMI-9589.

Fold	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
Fold1	0.9623	0.9637	90.86	92.49	89.04	90.37	81.66
Fold2	0.9574	0.9579	89.65	90.77	88.41	89.69	79.23
Fold3	0.9553	0.9513	90.03	91.24	88.70	89.97	80.00
Fold4	0.9631	0.9639	90.97	93.06	88.64	90.10	81.90
Fold5	0.9636	0.9611	91.23	92.61	89.70	90.90	82.41
Average	**0.9604**	**0.9596**	**90.55**	**92.03**	**88.90**	**90.21**	**81.04**
SD	0.0037	0.0052	0.67	0.98	0.50	0.46	1.36

Figure 4 presents the ROC and PR curves for CMI-9589. SMASH achieved a mean AUC of 0.9604 and a mean AUPR of 0.9596 under association-level five-fold cross-validation.

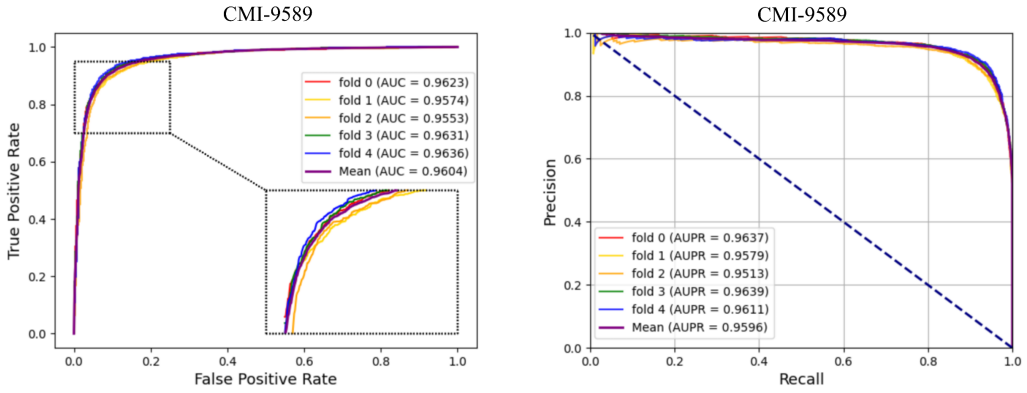

Figure 4: The ROC and PR curves for the CMI-9589 dataset.

F.3 CMI-20208 Dataset

Table 10 presents the five-fold cross-validation results for CMI-20208. The reported mean AUC and AUPR were 0.9572 and 0.9566, respectively. Fold-level AUC ranged from 0.9520 to 0.9649, while AUPR ranged from 0.9498 to 0.9660.

Table 10: Detailed five-fold cross-validation results on CMI-20208.

Fold	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
Fold1	0.9520	0.9498	89.84	92.01	87.43	89.05	79.63
Fold2	0.9566	0.9565	90.04	91.39	88.53	89.85	80.01
Fold3	0.9534	0.9508	89.79	91.54	87.84	89.33	79.52
Fold4	0.9595	0.9599	90.05	90.84	89.16	90.31	80.04
Fold5	0.9649	0.9660	90.77	91.82	89.61	90.76	81.49
Average	**0.9572**	**0.9566**	**90.10**	**91.52**	**88.51**	**89.86**	**80.14**
SD	0.0052	0.0067	0.39	0.45	0.90	0.70	0.79

Figure 5 presents the ROC and PR curves for CMI-20208. These evaluations report within-dataset performance under association-level five-fold cross-validation.

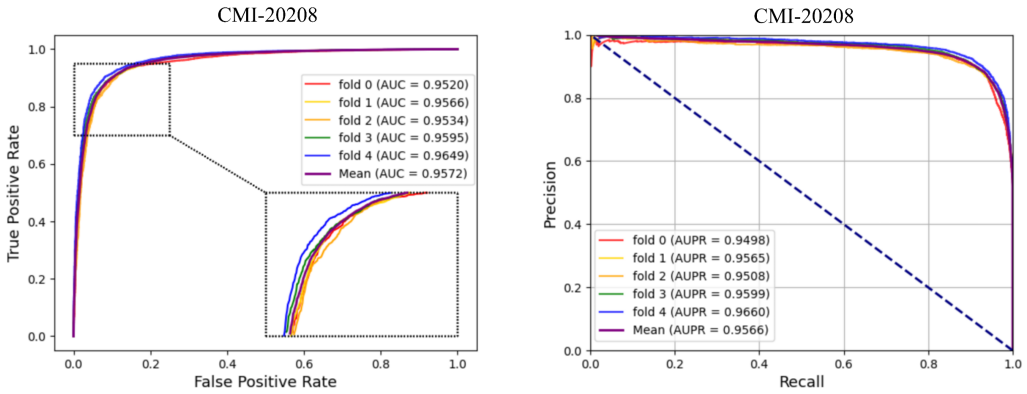

Figure 5: The ROC and PR curves for the CMI-20208 dataset.

G COLD-START EVALUATION

We evaluated RNAs with withheld interaction profiles under RNA-disjoint supervised splits, with their features available during representation learning. In the unseen-circRNA protocol, circRNAs were partitioned into five folds. For fold f, let $\mathcal{C}_{\text{test}}^{(f)}$ be the held-out circRNAs. The labeled training pairs satisfy

$$\mathcal{T}_f = \{(i,j) : (c_i, m_j, y_{ij}) \in \mathcal{D}, c_i \notin \mathcal{C}_{\text{test}}^{(f)}\}, \tag{G.1}$$

where $\mathcal{D}$ contains labeled positive and candidate-negative pairs. Both positive and negative pairs involving held-out circRNAs were excluded from supervised representation learning, CatBoost fitting, and its internal validation split. Test positives and IntaRNA-screened candidate negatives were restricted to pairs involving the held-out circRNAs. The unseen-miRNA protocol applied the corresponding restriction to held-out miRNAs.

The training matrix $Y^{(f)}$ retained only training positive associations. All rows corresponding to held-out circRNAs, or columns corresponding to held-out miRNAs, were set to zero. The joint operator was normalized before its RNA-specific principal submatrices were extracted. All model parameters were initialized within each fold. The standardizer was fitted on the training pair features.

Held-out RNAs remained in the node set. Their sequence, structural, and composition similarities were available during representation learning. They participated in adaptive graph construction and auxiliary masked reconstruction using the fold-specific inputs. Their held-out interaction labels were excluded from all training objectives. This is a transductive representation-learning setting with RNA-disjoint supervised labels. It evaluates interaction-profile withholding for a fixed node set. Insertion of new RNA nodes after model training requires a separate evaluation protocol. Both protocols used a 1:1 positive-to-negative ratio. Other hyperparameters follow Appendix C; metric definitions follow Appendix D. Reported values are means and standard deviations across five folds.

An RNA with no training interactions forms a singleton hyperedge in the initial joint construction. Its vertex degree and the degree of that hyperedge are both one. Consequently,

$$G_r^{\text{init}}[i,i] = 1, \qquad G_r^{\text{init}}[i,j] = 0 \quad (j \neq i). \tag{G.2}$$

Its initial propagation branch transforms its own features. Connections from the adaptive branch can change its effective propagation neighborhood. For a general incidence matrix, zero-degree inverse factors are set to zero using q_α; empty hyperedges then have zero contribution. The center-inclusive initial construction retains every hyperedge and requires no empty-hyperedge deletion. The adaptive incidence construction also includes each node on the diagonal. Inverse-degree factors are evaluated only at positive degrees, with zero assigned elsewhere.

Table 11: Cold-start performance under RNA-disjoint supervised five-fold splits, with held-out RNA features available during representation learning. Values are reported as mean ± standard deviation.

Dataset	Protocol	AUC	AUPR
CMI-9905	Unseen circRNA	0.8316 ± 0.0114	0.8372 ± 0.0117
CMI-9905	Unseen miRNA	0.7039 ± 0.0217	0.7246 ± 0.0214
CMI-9589	Unseen circRNA	0.8128 ± 0.0084	0.8127 ± 0.0060
CMI-9589	Unseen miRNA	0.7009 ± 0.0256	0.7170 ± 0.0243
CMI-20208	Unseen circRNA	0.8449 ± 0.0082	0.8437 ± 0.0089
CMI-20208	Unseen miRNA	0.7180 ± 0.0241	0.7274 ± 0.0202

Table 11 reports ranking performance for RNAs whose interaction profiles were withheld while their similarity features remained available. Under the unseen-circRNA protocol, AUC ranges from 0.8128 to 0.8449, and AUPR ranges from 0.8127 to 0.8437. Under the unseen-miRNA protocol, AUC ranges from 0.7009 to 0.7180, and AUPR ranges from 0.7170 to 0.7274. The lower unseen-miRNA scores characterize the performance of the complete model under this feature and propagation configuration. These results evaluate the complete model with held-out interaction profiles and available RNA features. They do not isolate the contribution of adaptive propagation.

H PARAMETER ANALYSIS

To determine the optimal parameter configurations of Multi-scale HGCN and GraphMAE and their influence on model performance, several parameter sensitivity analyses were performed. Experiments were conducted on varying scales of Multi-scale HGCN, different decoder types in GraphMAE, and diverse occlusion ratios to evaluate their respective effects on the performance of the SMASH model.

We varied N_{scales} from 1 to 5 to examine the influence of propagation depth. Table 12 and Figure 6 report the results. Increasing the number of scales from 1 to 3 raises AUC from 0.8938 to 0.9470 and AUPR from 0.8987 to 0.9487. Three scales provide the highest values for all reported metrics. At four scales, AUC and AUPR decrease to 0.9278 and 0.9293. At five scales, they decrease further to 0.9137 and 0.9188.

Table 13 and Figure 6 compare MLP, GCN, GIN, GAT, and HGCN decoders. The HGCN-based variant achieved the highest mean values across the seven reported metrics, with an AUC of 0.9470 and an AUPR of 0.9487. The GAT-based variant obtained the second-highest AUC and AUPR, at 0.9206 and 0.9218, respectively. The MLP-based variant obtained an AUC of 0.9142 and an AUPR of 0.9149. The corresponding values were 0.9102 and 0.9123 for GCN, and 0.8873 and 0.8956 for GIN.

Table 14 and Figure 6 report results for masking ratios of 0.1, 0.3, 0.5, 0.7, and 0.9. A masking ratio of 0.5 achieved the highest mean AUC and AUPR among the evaluated settings, at 0.9470 and 0.9487, respectively. At ratios of 0.1 and 0.3, AUC/AUPR values were 0.9207/0.9235 and 0.9357/0.9354. At ratios of 0.7 and 0.9, the corresponding values were 0.9282/0.9309 and 0.9151/0.9158.

Table 12: Performance analysis of different scales of Multi-scale HGCN.

Scale	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
1	0.8938 (±0.0051)	0.8987 (±0.0090)	82.83 (±0.86)	81.21 (±1.17)	84.44 (±0.71)	83.92 (±0.76)	65.69 (±1.70)
2	0.9276 (±0.0060)	0.9251 (±0.0063)	86.08 (±0.75)	84.95 (±1.21)	87.22 (±0.67)	86.92 (±0.64)	72.19 (±1.49)
3	**0.9470** (±0.0032)	**0.9487** (±0.0025)	**88.47** (±0.58)	**88.47** (±0.76)	**88.48** (±0.92)	**89.20** (±0.77)	**76.93** (±1.16)
4	0.9278 (±0.0037)	0.9293 (±0.0051)	85.74 (±0.70)	84.56 (±1.19)	87.01 (±0.57)	87.50 (±0.52)	71.54 (±1.39)
5	0.9137 (±0.0078)	0.9188 (±0.0096)	84.08 (±0.78)	81.64 (±1.28)	86.70 (±0.86)	86.84 (±0.76)	68.32 (±1.53)

Table 13: Performance analysis of different decoders in GraphMAE.

Decoder	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
MLP	0.9142 (±0.0071)	0.9149 (±0.0055)	84.24 (±0.59)	81.84 (±1.55)	86.64 (±0.50)	85.96 (±0.30)	68.56 (±1.11)
GCN	0.9102 (±0.0065)	0.9123 (±0.0073)	83.83 (±0.76)	81.24 (±1.16)	86.47 (±0.65)	85.97 (±0.66)	67.78 (±1.50)
GIN	0.8873 (±0.0060)	0.8956 (±0.0065)	82.22 (±0.48)	81.23 (±0.84)	83.22 (±0.18)	82.88 (±0.27)	64.47 (±0.94)
GAT	0.9206 (±0.0045)	0.9218 (±0.0044)	84.94 (±0.80)	83.01 (±1.61)	86.98 (±0.63)	87.04 (±0.53)	69.99 (±1.53)
HGCN	**0.9470** (±0.0032)	**0.9487** (±0.0025)	**88.47** (±0.58)	**88.47** (±0.76)	**88.48** (±0.92)	**89.20** (±0.77)	**76.93** (±1.16)

I COMPARISON OF GRAPH REPRESENTATION LEARNING

SMASH uses a GraphMAE-based branch to reconstruct masked backbone representations. We compare GAE (Zhang et al., 2024a), GAT (Zhang et al., 2024b), GATE (Luo & Deng, 2024), and

VGAE (Fan et al., 2025) as alternative graph representation modules while keeping the remaining pipeline and evaluation protocol fixed. Table 15 and Figure 7 report the results.

Table 15 compares the graph representation learning variants. The GraphMAE-based variant achieved the highest mean values across all seven metrics, with an AUC/AUPR of 0.9470/0.9487. These values were 0.0244/0.0238 higher than those of GAE, which ranked second on both metrics. The corresponding differences in accuracy, sensitivity, specificity, precision, and MCC were 3.05, 4.40, 1.61, 1.88, and 6.03 percentage points, respectively.

Table 14: Performance analysis of GraphMAE at different mask ratios.

Mask ratio	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
0.1	0.9207 (±0.0084)	0.9235 (±0.0070)	85.09 (±0.90)	83.24 (±1.84)	87.08 (±0.45)	87.39 (±0.38)	70.29 (±1.72)
0.3	0.9357 (±0.0059)	0.9354 (±0.0055)	86.97 (±0.75)	86.00 (±1.64)	88.02 (±0.50)	88.54 (±0.36)	73.99 (±1.45)
0.5	**0.9470** (±0.0032)	**0.9487** (±0.0025)	**88.47** (±0.58)	**88.47** (±0.76)	**88.48** (±0.92)	**89.20** (±0.77)	**76.93** (±1.16)
0.7	0.9282 (±0.0061)	0.9309 (±0.0064)	85.86 (±0.77)	85.24 (±1.04)	86.54 (±1.08)	87.20 (±0.90)	71.74 (±1.53)
0.9	0.9151 (±0.0094)	0.9158 (±0.0082)	84.62 (±1.15)	82.60 (±1.20)	86.74 (±1.33)	86.78 (±1.28)	69.34 (±2.31)

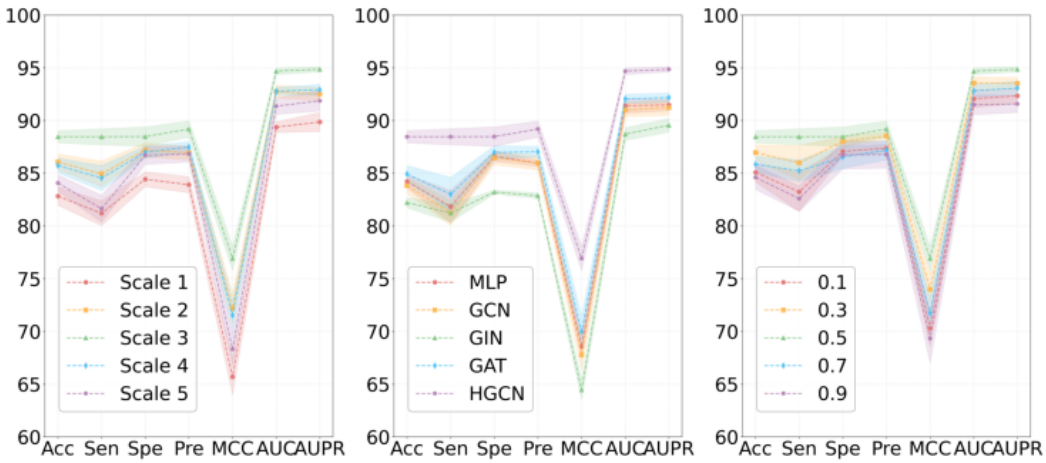

Figure 6: Sensitivity to HGCN propagation depth, GraphMAE decoder architecture, and masking ratio on CMI-9905.

Table 15: Comparison of graph representation learning variants on CMI-9905.

Model	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
GAE	0.9226 (±0.0045)	0.9249 (±0.0033)	85.42 (±0.63)	84.07 (±0.93)	86.87 (±0.52)	87.32 (±0.50)	70.90 (±1.23)
GAT	0.8845 (±0.0059)	0.8962 (±0.0058)	82.04 (±0.58)	79.31 (±1.26)	84.91 (±0.35)	84.70 (±0.26)	64.25 (±1.08)
GATE	0.9182 (±0.0032)	0.9231 (±0.0057)	84.60 (±0.55)	82.25 (±0.67)	87.12 (±0.87)	87.30 (±0.77)	69.35 (±1.12)
VGAE	0.9096 (±0.0071)	0.9113 (±0.0068)	83.62 (±0.81)	81.29 (±1.12)	85.99 (±1.08)	85.56 (±0.97)	67.34 (±1.61)
GraphMAE	**0.9470** (±0.0032)	**0.9487** (±0.0025)	**88.47** (±0.58)	**88.47** (±0.76)	**88.48** (±0.92)	**89.20** (±0.77)	**76.93** (±1.16)

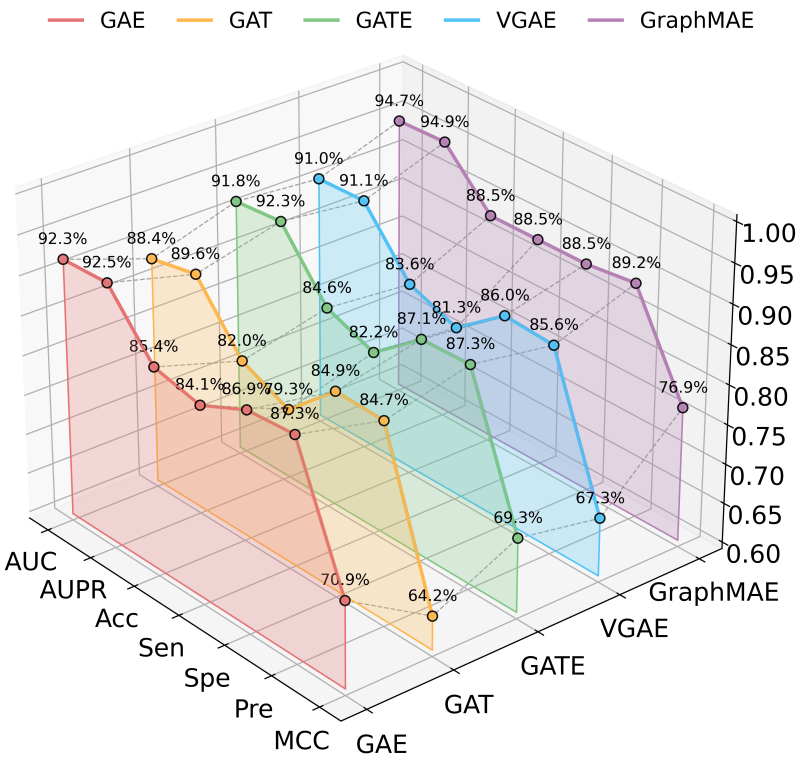

Figure 7: Comparison of graph representation learning variants on CMI-9905.

J Comparison of classification algorithms

To effectively exploit the complex relationships between the extracted circRNAs and miRNAs, the CatBoost classification algorithm was employed to predict potential CMIs. We conducted a comparative experiment with six widely used classification algorithms: XGBoost (Liang et al., 2024), Logistic Regression (LR) (Li et al., 2024), AdaBoost (Xuan et al., 2024), LightGBM (Zhang et al., 2025), Support Vector Machine (SVM) (Chen et al., 2024), and Gradient Boosting (GB) (Cui et al., 2025). During the comparison, only the classification algorithm module was replaced, while all other components of the framework remained unchanged. The comprehensive experimental results are presented in Table 16 and Figure 8. Table 16 shows that CatBoost achieved the highest mean AUC and AUPR among the evaluated classifiers, at 0.9470 and 0.9487, respectively. LightGBM obtained the second-highest values for these metrics, at 0.9237 and 0.9256. The corresponding differences were 0.0233 in AUC and 0.0231 in AUPR. CatBoost also achieved the highest mean accuracy, sensitivity, precision, and MCC. LightGBM achieved a specificity of 88.52%, compared with 88.48% for CatBoost.

K Extended Comparison with State-of-the-Art Models

We compare SMASH with published AUC and AUPR results from five CMI prediction methods across the three benchmark datasets. The models selected for comparison included BioDGW-CMI (Wang et al., 2023), BEROLECMI (Wang et al., 2024), MRHRL (Yin et al., 2024), WSCD (Guo et al., 2022), and CA-CMA (Guo et al., 2024), all evaluated using five-fold CV. The AUC and AUPR values of these baseline models were retrieved from their respective published studies. The comparative results for the CMI-9905, CMI-9589, and CMI-20208 datasets are presented in Tables 17, 18, and 19, respectively.

Table 16: Performance analysis of different classification algorithms.

Classifiers	AUC	AUPR	Acc(%)	Sen(%)	Spe(%)	Pre(%)	MCC(%)
XGBoost	0.9060 (±0.0057)	0.9084 (±0.0070)	83.66 (±0.61)	80.57 (±1.34)	86.75 (±0.89)	85.88 (±0.74)	67.45 (±1.19)
LR	0.8869 (±0.0057)	0.8923 (±0.0041)	82.34 (±0.52)	79.65 (±1.21)	85.04 (±0.61)	84.19 (±0.47)	64.79 (±1.01)
AdaBoost	0.8813 (±0.0050)	0.8894 (±0.0056)	81.53 (±0.71)	79.13 (±1.92)	83.93 (±0.73)	83.12 (±0.43)	63.15 (±1.36)
LightGBM	0.9237 (±0.0069)	0.9256 (±0.0050)	85.52 (±0.97)	82.51 (±1.73)	88.52 (±0.54)	87.78 (±0.61)	71.17 (±1.88)
SVM	0.8810 (±0.0050)	0.8860 (±0.0035)	81.66 (±0.59)	79.80 (±0.86)	83.53 (±1.08)	82.90 (±0.90)	63.38 (±1.19)
GB	0.9148 (±0.0065)	0.9176 (±0.0064)	84.43 (±1.00)	82.00 (±1.32)	86.86 (±0.83)	86.19 (±0.90)	68.95 (±1.97)
CatBoost	**0.9470** (±0.0032)	**0.9487** (±0.0025)	**88.47** (±0.58)	**88.47** (±0.76)	**88.48** (±0.92)	**89.20** (±0.77)	**76.93** (±1.16)

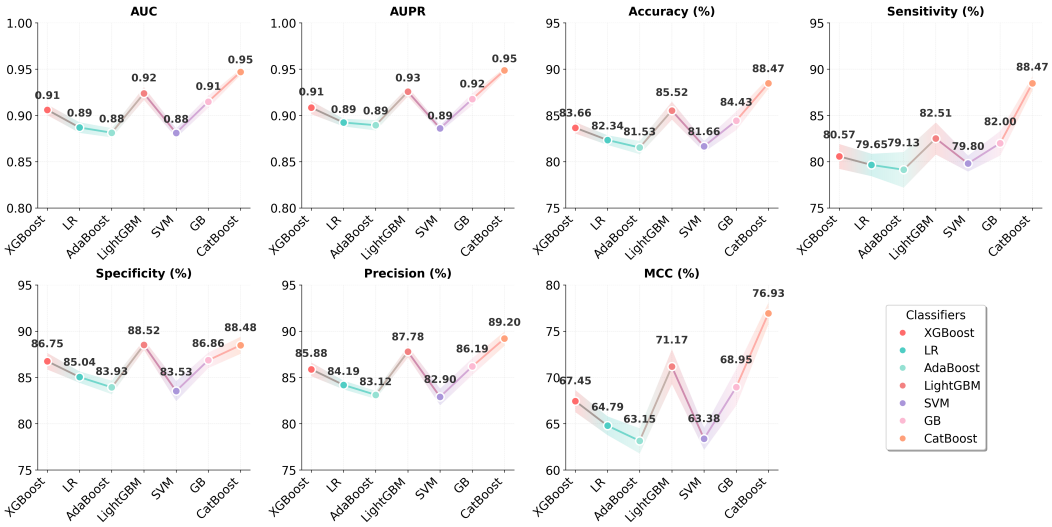

Figure 8: Comparison of classifiers using SMASH pairwise representations on CMI-9905.

Table 17: Comparison with published baseline results on CMI-9905.

Methods	BioDGW-CMI	BEROLECMI	MRHRL	SMASH
AUC	0.9026	0.9104	0.9276	**0.9470**
AUPR	0.8962	0.9086	0.9396	**0.9487**

Table 18: Comparison with published baseline results on CMI-9589.

Methods	MRHRL	BioDGW-CMI	BEROLECMI	SMASH
AUC	0.9404	0.9476	0.9491	**0.9604**
AUPR	0.9447	0.9416	0.9431	**0.9596**

Table 19: Comparison with published baseline results on CMI-20208.

Methods	WSCD	CA-CMA	MRHRL	SMASH
AUC	0.8898	0.9170	0.9264	**0.9572**
AUPR	0.8847	0.9131	0.9297	**0.9566**

Across the three datasets, the tabulated AUC and AUPR values for SMASH are numerically higher than the included published baseline values. On CMI-9905, the differences relative to MRHRL are 0.0194 in AUC and 0.0091 in AUPR. On CMI-9589, the corresponding differences are 0.0113 relative to BEROLECMI in AUC and 0.0149 relative to MRHRL in AUPR. On CMI-20208, the differences relative to MRHRL are 0.0308 in AUC and 0.0269 in AUPR.