

## A Training process

As in Figure 1, we plot the curves of average accuracy(AP) and model loss with the number of training rounds. The AP shows an upward trend with the number of training rounds. Except for the dataset other than corel5k, the other four datasets all stabilize after 75 training rounds in terms of the AP parameter, while corel5k stabilizes only after 150 training rounds. And the training loss of the model shows a decreasing trend with the number of training rounds, but compared with the first 150 rounds drop, the model loss in the last 50 rounds has a weaker impact on the performance evaluation index, so the optimal number of training rounds for our model should be set between 150 and 200 rounds.

## B Different missing percentages for views or labels

As shown in Figure 2, we compare the training results with different labels loss rates at 50% view loss rate and the learning results with different views loss rates at 50% labels loss rate, respectively. From the figure, it can be seen that all evaluation metrics show a decreasing trend as the labels loss rate and views loss rate increase. It is obvious that missing views have a greater impact on the classification task and the performance metrics decrease more significantly. One possible reason is that the absence of views results in fewer effective features, which reduces the model’s classification performance. However, the absence of labels can compensate for the effect of insufficient supervision due to the absence of labels by enabling cross-sample information exchange through the category relevance of the hierarchical contrastive learning module.

## C DatasetS

Table 1: Attributes of five common datasets

dataset	view	sample	category	train sample
corel5k	6	4999	260	3500
Pascal07	6	9963	20	6975
ESPGame	6	20770	268	14539
Iaprtc12	6	19627	291	13739
Mirflickr	6	25000	38	17500

## D Evaluation metrics

We select common DiMvMLC evaluation metrics, including: Average Precision (AP), Ranking Loss (RL), Adapted Area Under Curve (AUC), OneError (OE) and Coverage (Cov).

- Average Precision(AP): is used to evaluate the model’s ability to rank each label and prediction accuracy, i.e., the model’s ability to rank positive labels first and its accuracy in recognizing positive labels.
- Ranking Loss (RL): calculates the model’s ability to rank positive and negative labels, which calculates the probability that a positive label is ranked behind a negative label. The smaller the sorting loss, the stronger the model’s ability to sort positive labels, i.e., positive labels are more likely to be ranked in front of negative labels.
- Adapted Area Under Curve(AUC): indicates the ability of the model to randomly select positive and negative samples for differentiation, the closer the AUC value is to 1, the better the model performs. the closer the AUC is to 1, the better the model’s classification performance is.
- OneError (OE): evaluates whether the model can correctly predict the label with the highest confidence level on each sample. It calculates the proportion of the highest confidence label predicted by the model that is inconsistent with the true label. the smaller the OneError, the higher the accuracy of the model in predicting the highest confidence label.

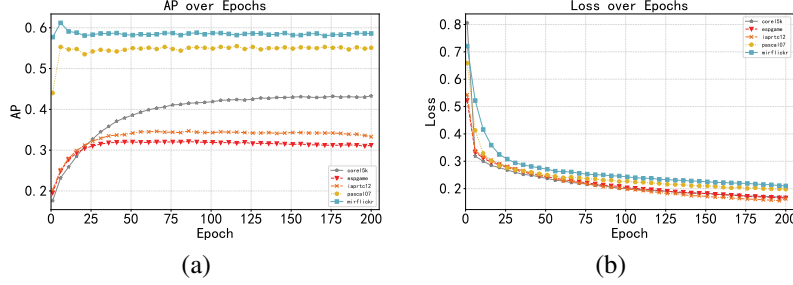


Figure 1: Variation of experimental parameters with the number of training rounds:(a) AP variation with epochs (b) Loss variation with epochs.

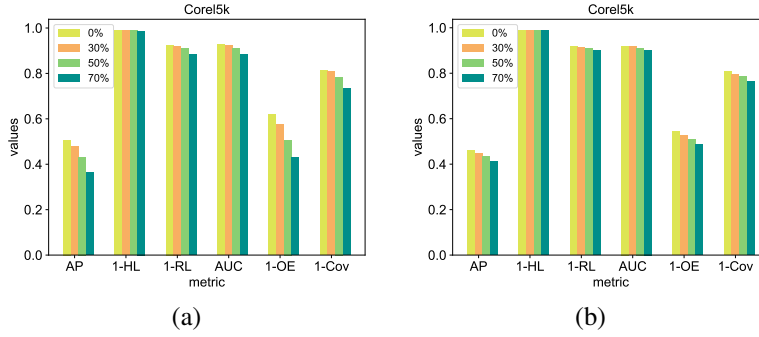


Figure 2: Results on the Core5k dataset with (a) different missing-view rates, (b) different missing-label rates.

- Coverage (Cov): this metric measures the proportion of the number of labels that need to be considered in order to cover all positive labels. It reflects the efficiency of the model in the recommendation task. The smaller the coverage, the lesser the number of labels the model needs to consider in the recommendation task and the more efficient it is.

## E Comparative learning

Contrastive Learning is a self-supervised learning method that aims to learn effective data representations by comparing pairs of positive and negative samples. Instead of relying on labels, this method uses structural information about the data itself to train the model. The core idea of Contrastive Learning is to keep similar data points close to each other in the representation space and keep dissimilar data points away from each other. Its general expression is:

$$-\log \frac{e^{\mathcal{S}(x, x^+)}}{e^{\mathcal{S}(x, x^+)} + \log \sum_{x^- \in \mathcal{F}(x)} e^{\mathcal{S}(x, x^-)}} \approx -\mathcal{S}(x, x^+) + \log \sum_{x^- \in \mathcal{F}(x)} e^{\mathcal{S}(xx^-)} \quad (1)$$

## F Computer configuration and social impact

Our model is programmed in python language and the model framework is built using pytorch. CentOS Linux 7.9 operating system is used in supercomputing platform and GPUs are utilized to obtain training models and validation results quickly.

The realization of this experiment promotes the development of machine learning field. By integrating multi-dimensional data and complex labeling relationships, this model promotes the practicalization of AI models in real scenarios, which can facilitate the development of computer vision technology in the fields of healthcare, multimedia, and automated driving with no adverse social impact.