Supplementary Material

A. Datasets

Two medical cross-modal datasets, i.e., Open-I and MIMIC-CXR, which are commonly used in Med-CMH tasks, are selected for our experiments. Additionally, we process all datasets to ensure that each pair contains at least one label. The details for each dataset are provided below:

Open-I [5] dataset boasts 2,818 pairs of X-ray images and corresponding radiology reports. In our experiments, we randomly select 500 pairs in Open-I as the query set, the remaining 2,318 pairs as the retrieval set, and 1,000 pairs sampled from the retrieval set to form the training set.

MIMIC-CXR [11] is a large-scale chest X-ray and radiology report dataset sourced from the Beth Israel Deaconess Medical Center between 2011 and 2016. Following the previous studies, we randomly select 2,000 pairs as the query set, the rest 87,286 pairs forming the retrieval set, and 10,000 pairs for training.

B. Training Pipline

As shown in Algorithm 1, the main objective of our work is to obtain unified noisy-free hash codes by mapping image and text data from a high-dimensional space into a common K-bit discrete Hamming space.

Algorithm 1: The training pipeline of our MCPH framework.

Input: A medical cross-modal dataset $\mathcal{D}$ with noisy correspondence;
Output: The hashing function and the unified noisy-free hash codes associated with image-text pairs;

Initialization: Obtain the image captions using the pre-trained LLM model CheXagent; Set all the hyper-parameters; Load the weight parameters of pre-trained CLIP (Frozen);
Prompt fine-tuning stage (Trainable):
for *each epoch* **do**
  for $i = 1 : num\_steps$ **do**
    Sample a mini-batch $\mathcal{D}_i = \{(\mathcal{I}_i, \mathcal{T}_i; \mathcal{C}_i; l_i)\}_{i=1}^{N}$, and enhance the model with Visual-Textual prompt;
    $[g_{i,j}^v, Z_{i,j}^v, \_] \leftarrow \text{Encoder}_\text{v}\big([g_{i,j-1}^v, Z_{i,j-1}^v, p_{i,j-1}^v]\big)$;
    $[g_{i,j}^t, Z_{i,j}^t, \_] \leftarrow \text{Encoder}_\text{t}\big([\overline{g}_{i,j-1}^t, Z_{i,j-1}^t, p_{i,j-1}^t]\big)$;
    Compute noise-robust contrastive learning loss $\mathcal{L}_{\text{nrcl}}$ in Eq. (3);
    Compute cross-modal hashing losses $\mathcal{L}_{\text{quan}}$, $\mathcal{L}_{\text{inter}}$, and $\mathcal{L}_{\text{intra}}$ in Eq. (6), Eq. (7), and Eq. (8);
    Utilizing L_{total} for optimizing the trainable parameters of MCPH;
    Generate $\boldsymbol{b}_i^{(*)} = \text{sign}(\lambda(h_i^v + h_i^t) + (1-\lambda)(f_i^v + f_i^t))$.
  end
end

C. Implementation Details

In our experiments, our MCPH framework is implemented based on Pytorch with a single RTX 3090 GPU. We adopt the CLIP architecture as our backbone and freeze all original parameters in the prompt-tuning stage. All images are resized to 224×224 for feature extraction. During the prompt-tuning stage, we optimize the trainable adapters and hashing layers on MIMIC-CXR dataset for 30 epochs and Open-I dataset for 200 epochs by using the AdamW optimizer with a weight decay of 0.01 and a learning rate of $5e-5$. In addition, the mini-batch size and *top-m* are set to 32 and 8, respectively. To mitigate information redundancy, we set k to 10 in Eq. (1) and Eq. (2) to integrate visual-textual prompts into our MCPH framework. Notably, we experimentally found that unfreezing the last two layers of the transformer-based encoder fully exploits the informative cues provided by prompts and enhances the overall learning capability. Furthermore, the hyperparameter λ in Eq. (5) is configured as 0.99 to balance global and local features during hash code learning. Meanwhile, the temperature τ is initially set to 0.07 and can be dynamically adjusted throughout the tuning process.

D. Additional Experiment Results

In this part, we conduct additional ablation studies to evaluate the impact of each component in MCPH. The comparative results as shown in Table 4, and five variations are involved, including a) **BASE** is regarded as the base model that only utilizes two transformer-based encoders of CLIP; b) **BASE+VPL** adds the VPL component based on the basic model; c) **BASE+TPL** adds the TPL component to the basic model, along with the image caption branch; d) **BASE+VPL+TPL** integrates the visual-textual prompt learning strategy into the base model; e) **MCPH** is considered as the "full" model. From the table, we can summarize that all designed modules can complement and reinforce each other, which further verifies the combined effects of our MCPH method.

Table 4. Ablation studies (%) on Open-I with 20% noise rate.

	Methods	mAP Scores			
		16bits	32bits	64bits	mean
I → T	BASE	54.56	55.22	55.58	55.12
	BASE+VPL	55.45	56.45	57.13	56.34
	BASE+TPL	56.74	56.71	58.78	57.41
	BASE+VPL+TPL	57.57	57.89	59.89	58.45
	MCPH	**58.24**	**59.13**	**60.44**	**59.27**
T → I	BASE	57.24	57.76	56.96	57.32
	BASE+VPL	60.34	61.13	59.93	60.47
	BASE+TPL	60.77	61.56	60.12	60.82
	BASE+VPL+TPL	61.39	62.68	61.63	61.90
	MCPH	**62.41**	**63.37**	**62.47**	**62.75**