Easy2Hard: From Partially to Fully Unmatched Modalities as Negative Samples in Contrastive Learning

Supplementary Material

7. Additional Experiment Details

7.1. Reproducibility Settings

All experiments on the MIMIC-series datasets and HoloAssist 5-modal dataset were performed on a high-performance computing cluster featuring NVIDIA L40S GPUs. Experiments on the Channel and MM-IMDb datasets were carried out on a separate workstation equipped with NVIDIA Quadro RTX 6000 GPUs.

7.2. Datasets and Splits

Symile-MIMIC. Symile-MIMIC is a clinical tri-modal dataset consisting of chest X-rays (CXR), electrocardiograms (ECG), and blood laboratory tests (labs), built from MIMIC-IV v2.2 (admissions and labs), MIMIC-IV-ECG v1.0 (ECGs), and MIMIC-CXR-JPG v2.0.0 (CXRs). Following Symile, each sample is constructed at the admission level: for a given admission, ECGs and labs are collected within the first 24 hours after admission, while CXRs are collected within a 24–72 hour window after admission; the earliest CXR, ECG, and labs in their respective windows are selected to form an aligned triplet. CXRs are restricted to PA/AP views and preprocessed by resizing the shorter side to 320 pixels, applying random square crops for training and center crops for validation/test, and normalizing with ImageNet mean and standard deviation. ECGs are 10-second, 12-lead recordings; samples containing NaNs or all-zero signals are removed, and ECG signals are normalized to $[-1, 1]$. Labs are represented using the 50 most common lab tests: values are percentile-normalized using the empirical CDF computed on the training set, concatenated with a 50-dimensional missingness indicator to form a 100-dimensional vector, and missing values are imputed with the corresponding training-set mean percentile. The dataset contains 11,622 admissions and follows the official patient-disjoint split lists released with Symile-MIMIC, with 95% of patients in the train/validation development split and 5% in the test split.

For evaluation, Symile considers zero-shot retrieval from (ECG, labs) to CXR using a 10-way candidate pool per query (one positive and nine randomly sampled negative CXRs from the test set).

EH-MIMIC. EH-MIMIC is a clinical multimodal dataset constructed from MIMIC-CXR-JPG v2.0.0 and MIMIC-IV v2.2, containing chest X-rays (CXR), radiology reports, and blood laboratory tests (labs). We first filter the MIMIC-CXR metadata to retain studies with valid acquisition dates and standard frontal views (AP/PA). For each retained study, we load the corresponding radiology report text (one report per study) and the associated CXR image.

To associate each CXR study with a hospital admission, we align studies to admissions at the patient level using the study acquisition date. Concretely, for a given subject, we identify candidate admissions whose hospitalization interval overlaps with the CXR study date (with a ± 1 day tolerance), and assign the study to the admission whose admission time is closest to the study date; the resulting admission identifier (`hadm_id`) is used to retrieve laboratory measurements. CXR images are preprocessed by resizing the shorter side to 320 pixels, applying a center crop to 320×320, and normalizing with ImageNet mean and standard deviation.

For labs, we consider numeric lab events (`valuenum`) from `labevents`. We select the 50 most frequent lab item IDs in the corpus and represent each admission as a 50-dimensional vector of raw lab values together with a corresponding 50-dimensional binary missingness mask, where each dimension indicates whether the lab was observed for that admission. Samples without available labs for the aligned `hadm_id` are discarded.

Finally, we perform patient-level data splitting by `subject_id` into train/validation/test sets with an 80%/10%/10% ratio to prevent patient leakage, ensuring that all studies from the same patient appear in only one split.

HoloAssist (5-modal). We use HoloAssist as a feasibility study to demonstrate applicability beyond 3 modalities. We use five modalities: eye gaze (`Eyes_sync.txt`), head pose (`Head_sync.txt`), and 3 IMU streams (`Accelerometer_sync.txt`, `Gyroscope_sync.txt`, `Magnetometer_sync.txt`).

For each recording, we align modalities by timestamps using the eye stream as the anchor: for every anchor timestamp, we select the nearest timestamp in each other modality, and keep the aligned time step only if the absolute time difference for *all* modalities is within a tolerance τ (we use $\tau = 0.02$s). Recordings with insufficient aligned points are discarded.

After strict alignment, we segment each recording into sliding windows of length 3 seconds with a stride of 1 second. Each window is kept only if it con-

tains at least 30 aligned time steps. For each modality within a window, we compute a fixed-length feature vector by concatenating per-dimension statistics: mean, standard deviation, minimum, and maximum over the aligned samples in that window. This yields one feature vector per modality per window, which we store as one JSONL entry. We follow the official HoloAssist split lists (`train-v1_2.txt`, `val-v1_2.txt`, `test-v1_2.txt`) to form train/validation/test sets, ensuring no recording appears in multiple splits.

7.3. Encoders and Input Preprocessing

Within each dataset, all compared methods share the same encoder families, initialization choices, and preprocessing so that only the loss construction and schedule differ. For the clinical trimodal datasets, modality-specific linear projections map encoder outputs to a shared space of dimension d=8192; for HoloAssist, we use d=1024.

Image Encoder. For image-based experiments (CXR), we use a ResNet-50 backbone. In the reported experiments, the image encoder is trained from scratch rather than initialized from pretrained weights. The final fully-connected layer is replaced with a linear projection to the shared embedding space of dimension d, followed by a LayerNorm on the projected embedding. CXR images are resized and cropped as described in the dataset preprocessing section and normalized using ImageNet mean and standard deviation.

Text Encoder. For text-based experiments, we use a Transformer-based text encoder following the Bio_ClinicalBERT architecture. In the main experiments, we initialize the text encoder from pretrained Bio_ClinicalBERT weights rather than training it from scratch. We tokenize the report text using the corresponding tokenizer and use the `[CLS]` representation from the last hidden layer as the sentence-level embedding (768-d). A linear layer projects this embedding to the shared d-dimensional space, followed by a LayerNorm. We adopt a pretrained text encoder because preliminary runs showed that training the text encoder from random initialization substantially degraded all compared objectives, shifting the difficulty from multimodal alignment to learning textual representations themselves.

ECG Encoder. For Symile-MIMIC experiments, we follow the encoder design used in Symile: a ResNet-18 backbone is used to encode 10-second 12-lead ECG signals, and its output is mapped to the shared d-dimensional embedding space via a linear projection. This encoder is trained from scratch in our experiments.

Tabular Encoder. For tabular features (labs), we use a three-layer MLP with GELU activations. This choice matches the low-dimensional structured nature of the laboratory inputs and avoids introducing method-specific architectural advantages. Specifically, the MLP maps the input lab vector to the shared d-dimensional embedding space with hidden sizes 256 and 1024, followed by a LayerNorm on the output embedding.

HoloAssist 5-modal Encoders. For the HoloAssist 5-modal feasibility study, each time window is represented by fixed-length vectors for five streams: `eyes`, `head`, `acc`, `gyro`, and `mag`. Because all five modalities are low-dimensional synchronized sensor streams rather than image or text inputs, we use lightweight per-modality MLP encoders in this setting.

Concretely, for each modality we compute per-dimension statistics (mean, standard deviation, minimum, and maximum) over aligned samples within a 3-second window, and concatenate them into a single vector; thus the input dimension for modality m is $4D_m$, where D_m is the number of columns in the corresponding `*_sync.txt` file after removing the timestamp column.

Each modality m is encoded by a lightweight two-layer MLP that maps the window feature vector to the shared embedding space

$$\mathbf{x}^{(m)} \in \mathbb{R}^{4D_m} \xrightarrow{\text{Linear}(4D_m \to 2d)} \text{ReLU} \xrightarrow{\text{Linear}(2d \to d)} \mathbf{z}^{(m)} \in \mathbb{R}^d.$$

We do *not* apply ℓ_2 normalization to encoder outputs in this setting, and we set the shared embedding dimension to d=1024.

Within the HoloAssist setting, all compared objectives (CLIP-Pairwise, Symile, Easy2Hard, and the ImageBind-like pivot baseline) share the same encoders and preprocessing; only the loss construction and/or schedule differs.

7.4. Training Recipe on HoloAssist

Optimizer and schedule. We train all HoloAssist models end-to-end with AdamW. Unless stated otherwise, we use a constant learning-rate schedule and train for 30 epochs.

Batching and compute. We use a training batch size of 2048 with gradient accumulation of 8 steps, and a validation batch size of 1024. All experiments are run on a single GPU.

Temperature scaling. We maintain a learnable scalar `logit_scale` for temperature scaling and enable exponential scaling (i.e., $\tau = \exp(\texttt{logit\_scale})$) in training and evaluation.

Checkpointing and selection. During training, we save checkpoints corresponding to the best validation retrieval accuracy and the best validation loss (`best_val_acc.pt` and `best_val_loss.pt`). Unless otherwise noted, we select checkpoints by validation Acc@1 for downstream test evaluation.

7.5. Hyperparameter Tuning and Model Selection

For HoloAssist, model selection is performed on the validation split while keeping encoder families, preprocessing, and optimization settings matched across objectives within each dataset. The detailed hyperparameter grid for the 5-modal HoloAssist study is given below.

Search protocol (HoloAssist 5-modal). For the HoloAssist feasibility study, we perform a grid search on the validation split while keeping encoders, preprocessing, batch construction, and optimization settings fixed across objectives. We sweep learning rate, weight decay, and the initialization of `logit_scale` for all methods, and additionally sweep the Easy2Hard sigmoid schedule parameters (midpoint and slope).

Hyperparameter candidates. For all objectives (Symile, CLIP-Pairwise, and ImageBind-like), we use a constant schedule and search over `lr` $\in \{5\times10^{-5}, 10^{-4}, 5\times10^{-4}\}$, `weight_decay` $\in \{10^{-3}, 10^{-2}, 10^{-1}\}$, and `logit_scale_init` $\in \{-7, -5, -3, -1\}$ (with exponential temperature scaling enabled). For Easy2Hard (sigmoid schedule), we use the same grid for (`lr`, `weight_decay`, `logit_scale_init`) and additionally search `midpoint` $\in \{9, 13, 15, 17, 21\}$ and `slope` $\in \{0.20, 0.25, 0.30, 0.35, 0.40\}$.

Evaluation and reporting. After selecting the best validation checkpoint for each configuration, we evaluate on the held-out test split under a fixed 10-way retrieval protocol (1 positive + 9 negatives per query), using the eye-gaze stream as the anchor modality. For uncertainty estimation, we report bootstrap confidence intervals with 10 bootstrap samples at $\alpha = 0.05$.

Within this grid, Easy2Hard attains the best test Acc@1 with `lr` $= 5 \times 10^{-5}$, `weight_decay` $= 10^{-3}$, `logit_scale_init` $= -7$, and sigmoid schedule parameters $(t_m, k) = (13, 0.2)$ (selected by validation Acc@1).

7.6. Results Beyond Three Modalities (M=5)

To evaluate scalability beyond trimodal settings, we conduct a 5-modal feasibility study on HoloAssist using `eyes`, `head`, `acc`, `gyro`, and `mag`. We use the predefined split lists released with HoloAssist (`train-v1_2.txt`, `val-v1_2.txt`, `test-v1_2.txt`) to build train/val/test JSONL files.

Hyperparameter	Candidates
Learning rate	$\{5\times10^{-5}, 10^{-4}, 5\times10^{-4}\}$
Weight decay	$\{10^{-3}, 10^{-2}, 10^{-1}\}$
logit_scale init	$\{-7, -5, -3, -1\}$
Easy2Hard midpoint	$\{9, 13, 15, 17, 21\}$
Easy2Hard slope	$\{0.20, 0.25, 0.30, 0.35, 0.40\}$

Table 6. Hyperparameter grids for HoloAssist 5-modal tuning. All objectives share the same grid for optimizer and temperature parameters; Easy2Hard additionally tunes the sigmoid schedule.

Protocol. We evaluate 10-way zero-shot retrieval on the test split using the eye stream as the anchor modality. For each query window, we sample one matched positive and $K=9$ negatives uniformly from the test candidate pool (excluding the positive) and report Acc@1. Checkpoints are selected by validation Acc@1, and uncertainty is estimated with 10 bootstrap samples ($\alpha=0.05$).

Because HoloAssist contains many windows per recording, negatives drawn from the global test pool may be easier than within-recording confusions due to recording/session cues. Still, all methods follow the same protocol, so this experiment serves as a feasibility check of whether Easy2Hard remains beneficial when scaling to $M=5$ modalities.