A Spatial Feature Extraction

We initially utilize a CRAFT OCR [4] to outline the text area (TA) of the frame F_d. Then we adopt the prompt encoder heuristic in the Segment Anything Model (SAM) [18] to convert TA into prompt embeddings X_{vt}^d to obtain the feature vector of the location and content information of the text area. At the same time, we input F_d into the pre-trained ViT to generate the initial encoding X_v^d to capture the spatial features of the image in a single frame.

For the obtained X_{vt}^d and X_v^d, we optimize the representation of visual features X_v^d through a bidirectional attention mechanism, making it more focused on the relative representation of subtitle features in space. The process mainly takes the following steps:

We input X_{vt}^d and X_v^d into a bidirectional attention block. The bidirectional attention block first performs a self-attention (Self-Att) operation on X_{vt}^d to enhance its own feature representation.

$$\hat{X}_{vt}^d = \frac{(X_{vt}^d + \text{Self-Att}(X_{vt}^d)) - \mu}{\sqrt{\sigma^2 + \epsilon}} \sim \text{NORM}(X_{vt}^d + \text{Self-Att}(X_{vt}^d)) \tag{16}$$

where Self-Att is self-attention, and we normalize all features within the sample to have zero mean and unit variance to help alleviate the problem of internal covariate shift. Later, we use **NORM** to simplify this normalization process. Then, we perform cross-attention (Cross-Att) operations on the optimized features $\hat{X}_{vt}^d$ and X_v^d to interact with each other and further optimize the $\hat{X}_{vt}^d$.

$$\dot{X}_{vt}^d = \text{NORM}(\hat{X}_{vt}^d + \text{Cross-Att}(\hat{X}_{vt}^d, X_v^d)) \tag{17}$$

Lastly, a cross-attention operation is performed on X_v^d to make it interact with the updated $\dot{X}_{vt}^d$ from MLP to obtain a more accurate visual feature representation.

$$\hat{X}_v^d = \text{NORM}(X_v^d + \text{Cross-Att}(\text{MLP}(\dot{X}_{vt}^d), X_v^d)) \tag{18}$$

After the attention process, the updated $\hat{X}_v^d$ is down-sampled through two convolutional layers and then expanded to obtain the spatial pattern features of a single frame X^d. Finally, we use self-attention to fuse these single-frame spatial features and obtain a unified spatial feature representation through average pooling (MEAN).

$$X_{spa} = \text{MEAN}(\text{Self-Att}([X^1, X^2, ..., X^d]) \tag{19}$$

B Domain datasets

Table 5: Domains division of FakeSV and FakeTT datasets.

Domain	FakeSV	FakeTT
Culture	278	325
Disaster	1836	182
Education	195	28
Finance	116	9
Health	841	121
Politics	296	715
Military	121	84
Science	234	271
Society	1621	337

We use the Qwen2.5-72B [41] combined with manual verification for label classification based on the text and fakesv's video collection introduction [28]. We divide the above two datasets into nine domains, respectively. Specific domains' names and numbers are detailed and shown in Table 5.

The specific prompt template is: *"Here are some fields: Society, Health, Disaster, Culture, Education, Finance, Politics, Science, Military. And some examples from different domains* ***[Some examples (Society: [case text], Health: [case text], ...)]****. You can then compare and analyze the* ***[input text]*** *with the examples above to choose the most suitable Domain."*

We divide the above two datasets into nine domains, respectively. Specific domains' names and numbers are detailed and shown in Table 5. Since the number of some domains is small (Education, Finance, and Military in the FakeTT dataset), we conduct merging experiments with them.

C More Experimental Analyses

C.1 Training Strategy

We define the total loss function $Loss_{\text{final}}$ as the combination of the **Cross-modal Interpolation Distillation** loss $Loss_{\text{dis}}$ and the **Cross-modal Invariance Fusion** loss $Loss_{\text{es}}$ (emotion and sentiment features) and $Loss_{\text{st}}$ (spatial and temporal features). The specifically $Loss_{\text{final}}$ we define is as follows:

$$Loss_{\text{final}} = \alpha Loss_{\text{es}} + \beta Loss_{\text{st}} + \gamma Loss_{\text{dis}}, \tag{20}$$

where α, β, and γ denote the weighting factors.

C.2 Experimental Implementations

Our experiments include two aspects: domain generalization experiments and general domain experiments. The comparison methods selected in the domain generalization experiments are mainly the best performing and latest methods selected from general domain experiments for comparison. All our experiments were conducted on an NVIDIA RTX A6000 (48G). The α in the loss functions $Loss_{final}$ is 0.1, β is 3, and γ is 0.05. We choose the learning rates of 5e-5 for FakeSV and 1e-3 for FakeTT.

For more details on the DOCTOR model, our anonymous source code is available at https://anonymous.4open.science/r/DOCTOR-D1D2.

C.3 Comparison of Domain Generalization Star Map

As shown in Figure 6, the experiment present a comparative analysis of six misinformation detection methods (FakingRecipe, SV-FEND, OpEvFake, CMRF*, MMDFND*, and DOCTOR) across two distinct datasets: FakeSV and FakeTT. The radar visualization effectively illustrates each method's performance across multiple content categories, including Disaster, Society, Health, Culture, Politics, Science, Education, Finance, and Military for FakeSV, with a combined Education/Finance/Military category for FakeTT. The charts reveal DOCTOR consistently outperforming other methods across most categories in both datasets, particularly excelling in Military (93.65%) for FakeSV and Culture (88.31%) for FakeTT.

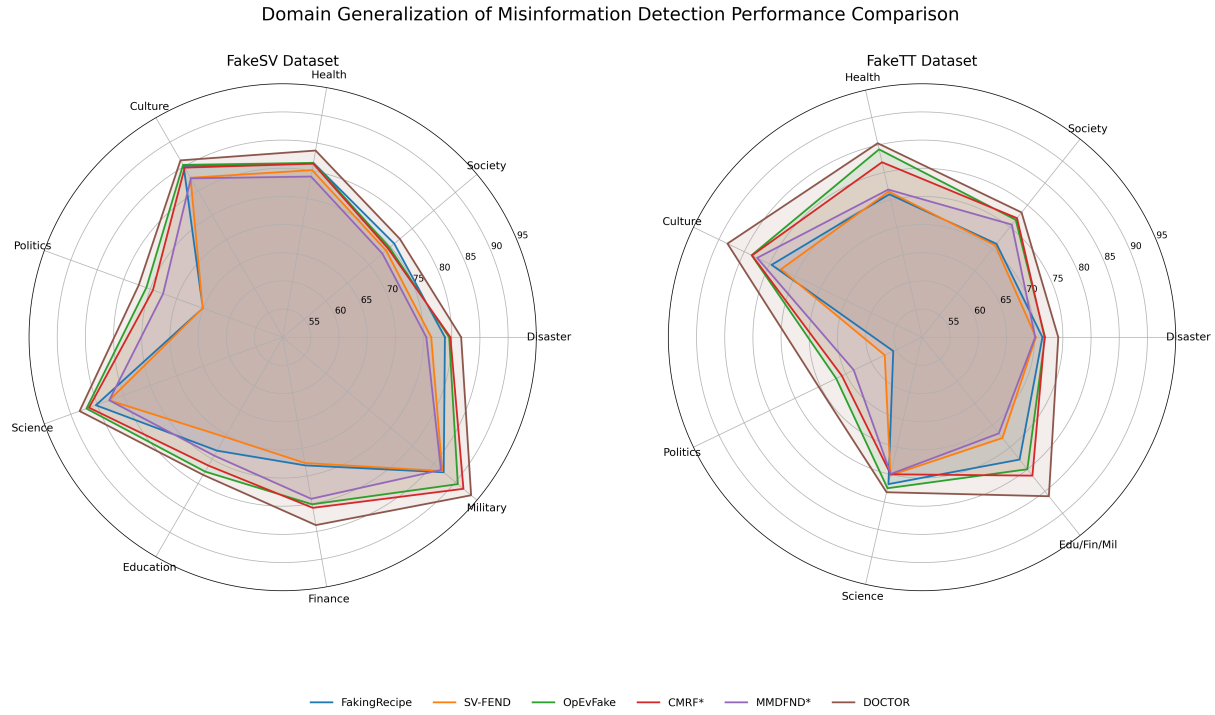

Figure 6: Comparative analysis of six misinformation detection methods (FakingRecipe, SV-FEND, OpEvFake, CMRF*, MMDFND*, and DOCTOR) across two distinct datasets: FakeSV and FakeTT. * represents the domain generalization method.

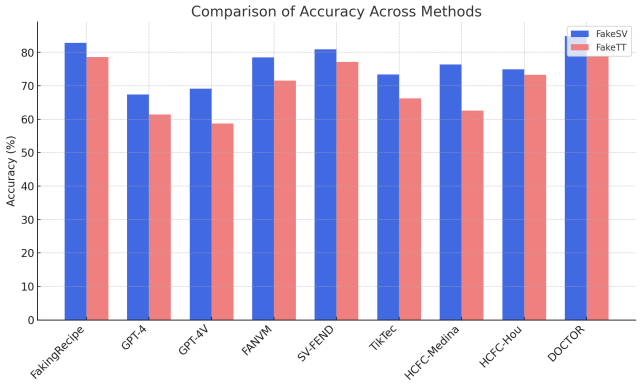

Figure 7: Accuracy comparison of different methods on FakeSV and FakeTT datasets.

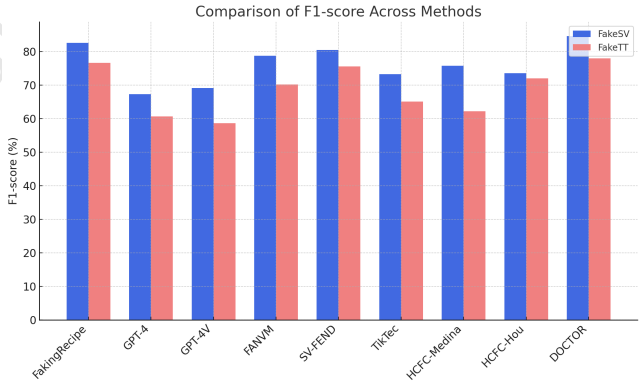

Figure 8: F1 comparison of different methods on FakeSV and FakeTT datasets.

C.4 More comparative accuracy analysis

The above two figures of Figure 7 and Figure 8 show the comparison of the Accuracy and F1-score of different methods on FakeSV and FakeTT datasets. It can be seen that the DOCTOR method achieved the best performance on both datasets.

The above three figures of Figure 11, Figure 10, and Figure 9 show the comparison of the Precision, Recall, and F1-score of different methods on the Fake category on the FakeSV and FakeTT datasets. It can be seen that the DOCTOR method performs outstandingly in Recall and F1-score on the Fake category.

The above three figures of Figure 14, Figure 13, and Figure 12 show the comparison of the Precision, Recall, and F1-score of the Real category of different methods on the FakeSV and FakeTT datasets. From the results, the DOCTOR method performs best in Precision while maintaining a high level in Recall and F1-score, and the overall performance is stable.

C.5 Discussion on DOCTOR Model Efficiency

Despite the advanced features and complex mechanisms integrated into the DOCTOR model, as shown in Figure 15, our experiments

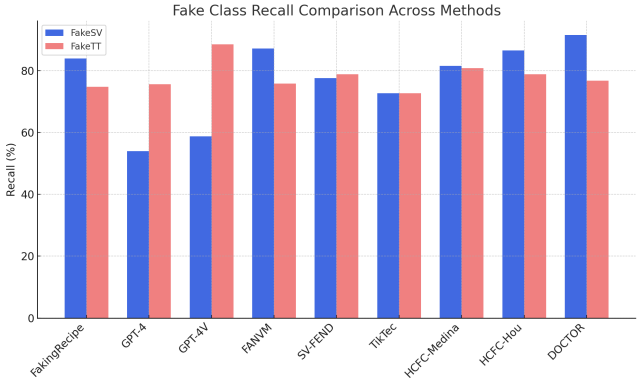

Figure 9: Recall comparison of different methods on FakeSV and FakeTT datasets (Fake class).

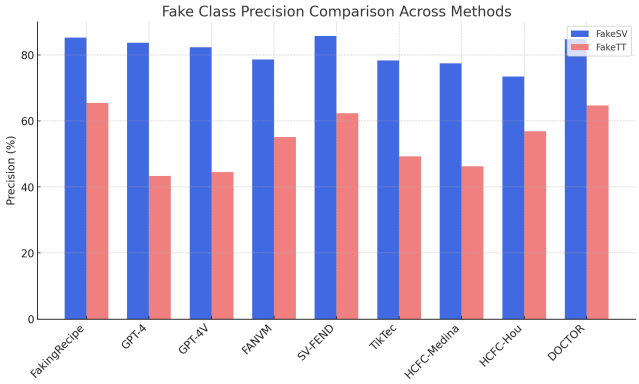

Figure 10: Precision comparison of different methods on FakeSV and FakeTT datasets (Fake class).

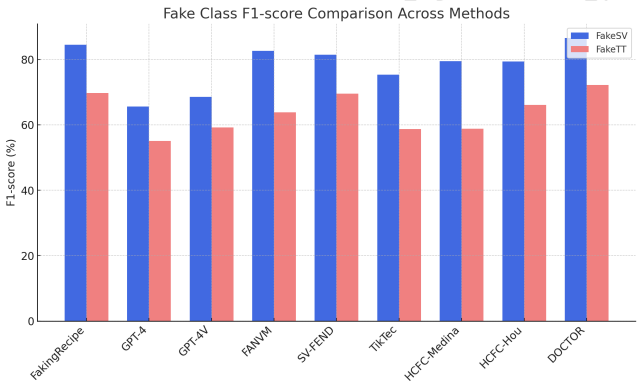

Figure 11: F1 comparison of different methods on FakeSV and FakeTT datasets (Fake class).

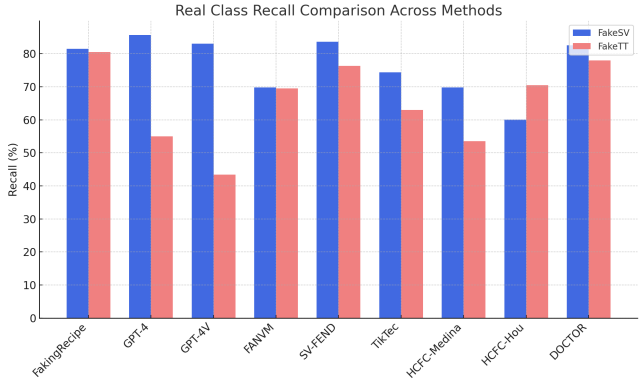

Figure 12: Recall comparison of different methods on FakeSV and FakeTT datasets (Real class).

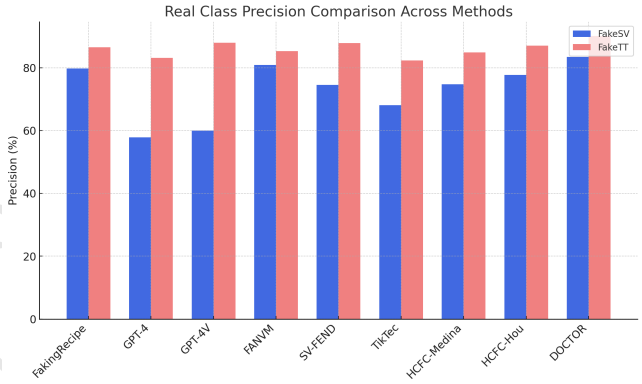

Figure 13: Precision comparison of different methods on FakeSV and FakeTT datasets (Real class).

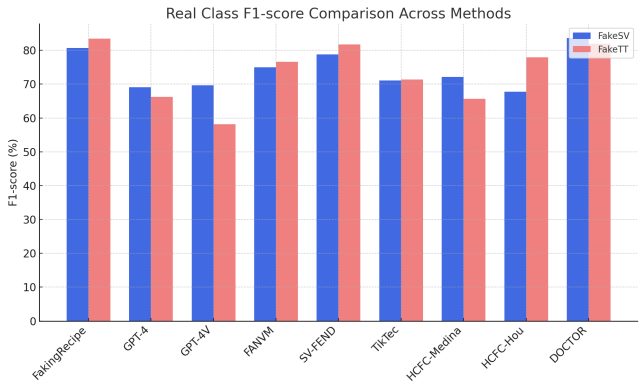

Figure 14: F1 comparison of different methods on FakeSV and FakeTT datasets (Real class).

reveal that the computational overhead does not experience a significant increase. The incorporation of cross-modal interpolation distillation and multi-modal invariance fusion, while enhancing the model's generalization ability across different domains, does not lead to a substantial rise in computational cost. This indicates that DOCTOR is able to maintain efficient performance, even with the addition of multiple modalities and the complexity of noise-based feature refinement. The efficiency of DOCTOR is crucial for real-world

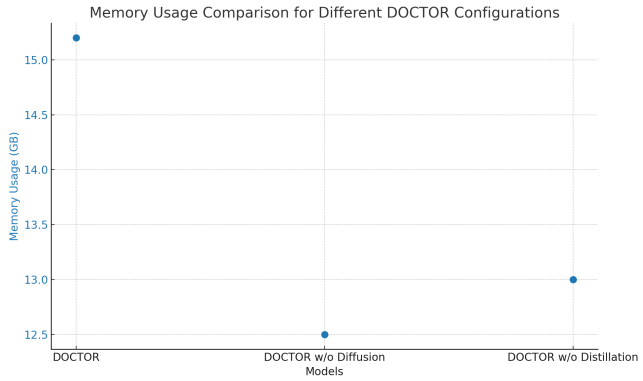

Figure 15: Discussion on DOCTOR Model Efficiency.

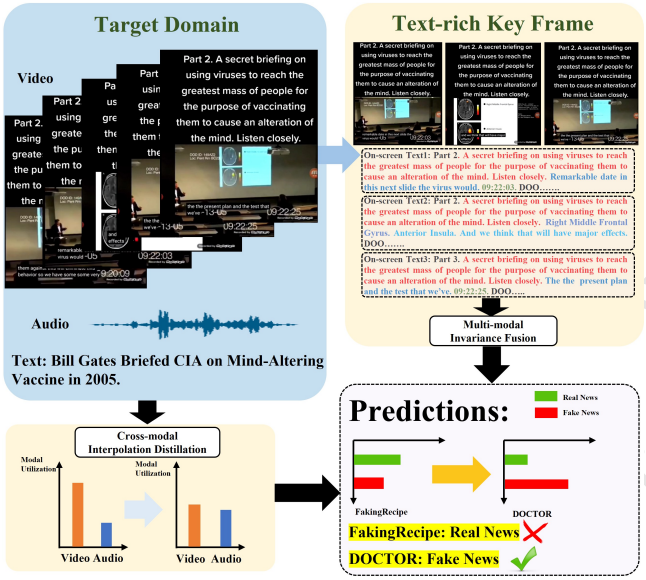

Figure 16: Case: Bill Gates Briefed CIA on Mind-Altering Vaccine in 2005.

applications, where both accuracy and computational efficiency are paramount.

C.6 Case Study

As shown in Figure 16. Case: Bill Gates Briefed CIA on Mind-Altering Vaccine in 2005.

In this case, the DOCTOR model demonstrates strong domain generalization capabilities by correctly identifying the video as misinformation, while traditional methods like FakingRecipe misclassify it as truth. They struggle to adapt when encountering unseen distributions, variations in video styles, or novel linguistic manipulations. In contrast, DOCTOR incorporates multi-modal invariance fusion and cross-modal interpolation distillation, allowing it to generalize across different domains by capturing latent relationships between text, audio, and video. This enables it to detect cross-modal inconsistencies, making it more robust against out-of-domain misinformation scenarios.

D Temporal Position Encoding

Temporal Position Encoding (TPE) is designed to capture the coarse temporal location of each segment within the entire video. To achieve this, we divide the video into B equal-frequency temporal bins and assign each segment to one of them based on its position in the sequence. Each bin is associated with a learnable embedding vector, which is used to represent the temporal context of segments assigned to that bin. This encoding helps the model understand whether a segment is located at the beginning, middle, or end of the video, which can be crucial for modeling temporal editing patterns.

Algorithm 2 Temporal Position Encoding (TPE)

Require: Number of segments n, number of bins B, embedding dimension d
Ensure: Temporal position embeddings $\{TPE^1, \ldots, TPE^n\}$

$s \leftarrow \lfloor \frac{n}{B} \rfloor$
Initialize embedding matrix $E \in \mathbb{R}^{B \times d}$
for $i = 1$ **to** n **do**
  $b_i \leftarrow \min\left(\lfloor \frac{i-1}{s} \rfloor, B-1\right)$
  $TPE^i \leftarrow E[b_i]$
end for
return $\{TPE^1, \ldots, TPE^n\}$

E Duration Encoding

Duration Encoding (DE) encodes the fine-grained temporal position of each segment using a fixed sinusoidal scheme, similar to the original positional encoding used in Transformers. This method ensures that each segment has a unique continuous representation based on its position index. For each segment i, and for each embedding dimension j, the encoding alternates between sine and cosine functions of different frequencies. This encoding provides precise temporal cues, enabling the model to capture temporal dependencies and alignment across modalities.

Algorithm 3 Duration Encoding (DE)

Require: Number of segments n, encoding dimension d
Ensure: Duration encodings $\{DE^1, \ldots, DE^n\}$

Initialize $DE \in \mathbb{R}^{n \times d}$ with zeros
for $i = 0$ **to** $n-1$ **do**
  for $j = 0$ **to** $d-1$ **do**
    $\theta \leftarrow \frac{i}{10000^{\frac{2\cdot\lfloor j/2 \rfloor}{d}}}$
    if j is even **then**
      $DE^{i+1}[j] \leftarrow \sin(\theta)$
    else
      $DE^{i+1}[j] \leftarrow \cos(\theta)$
    end if
  end for
end for
return $\{DE^1, \ldots, DE^n\}$

F Predication

Algorithm 4 Prediction via Feature Diffusion and Distillation Fusion

Input: Time feature X_{time}, Spatial feature X_{spa}, Semantic feature X_{sem}, Emotional feature X_{emo}, Video feature X_{video}, Audio feature X_{audio}
Output: Final prediction score $\hat{Y}_{\text{final}}$
{**Step 1: Feature fusion and diffusion**}
$X_{\text{all}} \leftarrow \text{Concatenate}(X_{\text{time}}, X_{\text{spa}}, X_{\text{sem}}, X_{\text{emo}})$
$X_{\text{diffused}} \leftarrow \text{ApplyDiffusion}(X_{\text{all}})$
{**Step 2: Time and space prediction branch**}
$X_{\text{ts}} \leftarrow \text{Extract}(X_{\text{diffused}}, [\text{time}, \text{spa}])$
$\hat{Y}_{\text{st}} \leftarrow \text{MLP}_{\text{TS}}(X_{\text{ts}})$
{**Step 3: Semantic and emotional prediction branch**}
$X_{\text{se}} \leftarrow \text{Extract}(X_{\text{diffused}}, [\text{sem}, \text{emo}])$
$\hat{Y}_{\text{es}} \leftarrow \text{MLP}_{\text{SE}}(X_{\text{se}})$
{**Step 4: Video-Audio distillation output**}
$\hat{Y}_{\text{dis}} \leftarrow \text{Distill}(X_{\text{video}}, X_{\text{audio}})$
{**Step 5: Late fusion to get final prediction**}
$\hat{Y}_{\text{fusion}} \leftarrow \hat{Y}_{\text{st}} \times \tanh(\hat{Y}_{\text{es}})$
$\hat{Y}_{\text{final}} \leftarrow \hat{Y}_{\text{fusion}} \times \hat{Y}_{\text{dis}}$
return $\hat{Y}_{\text{final}}$

To generate the final prediction, we design a multi-branch architecture that integrates temporal, spatial, semantic, and emotional features through a diffusion-enhanced fusion mechanism. As illustrated in Algorithm 4, the input features X_{time}, X_{spa}, X_{sem}, and X_{emo} are first concatenated and passed through a diffusion process to enhance inter-feature dependencies and latent structure alignment. The diffused representation X_{diffused} is then split into two branches:

Time-Spatial Branch: We extract the time and spatial-related features to form X_{ts}, which is passed through a dedicated MLP_{TS} to yield an intermediate prediction $\hat{Y}$st.

Semantic-Emotional Branch: Similarly, semantic and emotional components are extracted as X_{se}, which is processed via another MLP_{SE} to produce $\hat{Y}$es.

In parallel, we distill information from both video and audio modalities using a specialized distillation module, resulting in $\hat{Y}_{\text{dis}}$.

For final prediction, we adopt a late fusion strategy where the time-spatial output is modulated by the nonlinear activation of the semantic-emotional output: $\hat{Y}\text{fusion} = \hat{Y}\text{st} \cdot \tanh(\hat{Y}\text{es})$. This fusion is then further modulated by the distilled multi-modal prediction: $\hat{Y}\text{final} = \hat{Y}\text{fusion} \cdot \hat{Y}\text{dis}$. This design effectively captures complementary information from all feature domains and ensures robustness through multi-modal consistency.

For specific code implementation, please refer to the code link.