Wasserstein Modality Alignment Makes Your Multimodal Transformer More Robust

Zhuo Zhi [1] **Ziquan Liu** [2] **Qiangqiang Wu** [3] **Miguel Rodrigues** [1]

Abstract

Early fusion at a one-tower model such as a multimodal transformer is an effective multimodal learning paradigm. However, in a multimodal transformer, the modality fusion is performed solely through the self-attention function, which is originally designed for unimodal token sequences. To improve the self-attention mechanism for handling multimodal input, a parametric adapter model, like the Q-former in BLIP-2, is often used to align tokens from different modalities. Unlike existing methods that use an adapter model for modality alignment, our paper proposes an implicit approach based on Wasserstein distance that aligns tokens from different modalities in a multimodal transformer without using any additional parameters. Our empirical study shows that the implicit modality alignment improves the effectiveness of the multimodal Transformer in discriminative tasks, as well as its robustness to input noise and missing modalities. We conduct experiments on four different types of downstream task datasets, including both 2-modalities and 3-modalities tasks. In standard testing, testing with modality noise, and testing with missing modalities, the averaged improvement of our method compared with the baseline over all datasets are 0.9%, 2.5%, and 2.1% respectively.

1. Introduction

Multimodal machine learning (MML) mimics human perception by integrating multiple modalities such as text, audio, images, video, and sensor data to form a comprehensive understanding of the world. Many multimodal models have been applied to common tasks like multimodal medical diagnostics (Hayat et al., 2022), sentiment analysis (Zadeh et al., 2018) and malicious speech detection (Kiela et al., 2020).

Aligning heterogeneous data in multimodal learning is crucial since such data often exhibit varied distributions, representations, and noise levels. Proper alignment enhances the uniform representation of these diverse data types, leading to improved performance and robustness in multimodal tasks (Ghahremani Boozandani & Wachinger, 2024; Liang et al., 2024; Kim et al., 2020). To achieve better modality alignment, various strategies are applied in large-scale multimodal models, such as the Q-Former in BLIP-2 (Li et al., 2023), contrastive learning in CLIP (Radford et al., 2021) and Imagebind (Girdhar et al., 2023).

Multimodal models based on a one-tower transformer (multimodal transformer), by its flexibility and simplicity, are widely used for a variety of multimodal learning tasks (Lee et al., 2023; Nagrani et al., 2021; Zhi et al., 2024; Ma et al., 2021). Although the multimodal transformer can handle multimodal tokens as the input due to the flexibility of self-attention layers, it lacks a mechanism for modality alignment during the fine-tuning process. In other words, it is not optimal to rely on a pre-trained multimodal transformer to achieve the modal alignment (Kim et al., 2021; Wang et al., 2021). Taking ViLT (Kim et al., 2021) as an example, tokens from different modalities are concatenated together and processed by the multimodal transformer. Several learning tasks, such as image text matching and word patch alignment are applied during the pre-training phase to ensure the modality alignment. However, the alignment process is not enforced in the fine-tuning stage if the input tokens are from multimodal sources, which leads to deteriorated performance in a downstream task.

To address this issue, we propose Wasserstein Modality Alignment (WMA), an implicit regularization method to align the Wasserstein distance between two modalities within a multimodal transformer, as shown in Fig. 1. To make the computation feasible, we use the popular optimal transport (OT) distance (Peyré et al., 2019) as the instantiation of Wasserstein distance. We regularize the degree of modality alignment by adjusting the OT distance between

[1]Department of Electronic and Electrical Engineering, University College London, London, UK [2]School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK [3]Department of Computer Science, City University of Hong Kong, Hong Kong, China. Correspondence to: Zhuo Zhi <zhuo.zhi.21@ucl.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

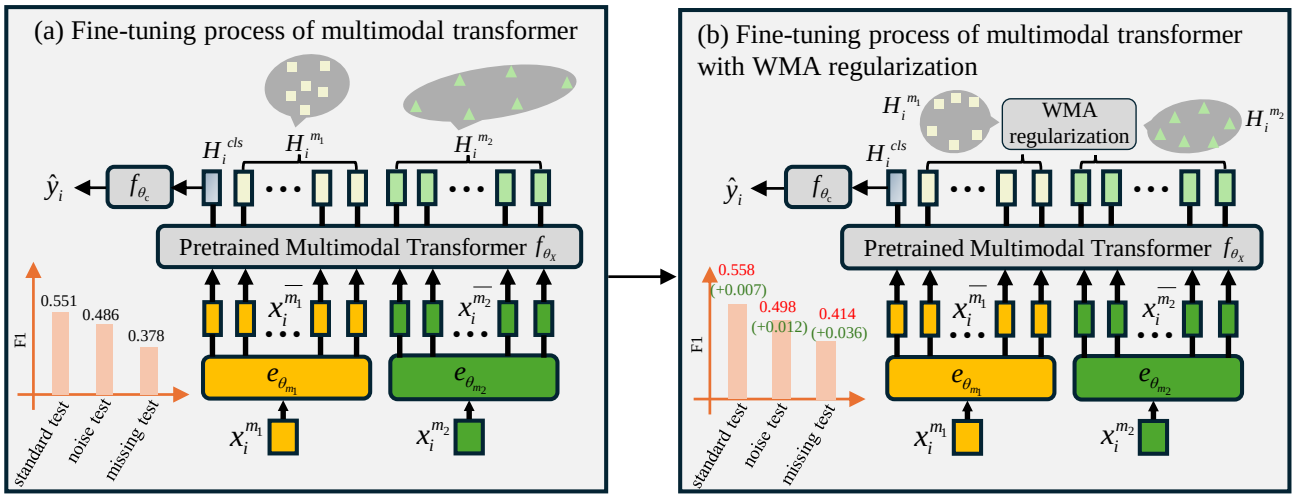

Figure 1. The overview of the proposed method. (**a**) Multi-label classification of movies by fine-tuning ViLT model on MM-IMDb dataset. The image data $x_i^{m_1}$ and text data $x_i^{m_2}$ are firstly processed by the embedding operation $e_{\theta_{m_1}}$ and $e_{\theta_{m_2}}$ to get the token sequence $x_i^{\bar{m}_1}$ and $x_i^{\bar{m}_2}$, which then are concatenated and processed by the multimodal transformer f_{θ_X}. $H_i{}^{cls}$ in the multimodal transformer is input into the classifier f_{θ_c} for predicting the label $\hat{y}_i$. Note that there is no specific module to adjust modality alignment in (a). (**b**) We propose the WMA regularization method to search for the optimal modality alignment for the target task. WMA calculates the OT distance between $H_i{}^{m_1}$ and $H_i{}^{m_2}$ to represent the alignment degree of two modalities. The target OT distance range is set to search for the optimal alignment. Our proposed WMA method effectively improves the performance and robustness of the model, as shown in the lower left corner of (**1**) and (**2**) (the experiment results on MM-IMDb datasets, 'noise test' refers to the test with noise on text and 'missing test' refers to the test with missing text. See more details in Table. 1)

different modalities' feature distributions. Interestingly, our empirical study demonstrates that directly minimizing the OT distance between two modalities often leads to inferior performance, so our WMA aligns two modalities with a task-dependent modality distance. In the practical sense, the proposed WMA is a plug-and-play method and does not introduce any additional training parameters. Our main contributions are three-fold:

- We propose to perform the modality alignment in the fine-tuning process of a pre-trained multimodal transformer without any additional adapter and design the Wasserstein Modality Alignment based on the optimal transport distance to achieve lightweight modality alignment in the feature tokens of the transformer.
- Instead of minimizing the OT distance between any two modalities, our WMA is a task-dependent modality alignment method that can handle different requirements for the degree of modality alignment.
- We evaluate our proposed WMA on four datasets, including 2-modalities and 3-modalities tasks. Our experimental results demonstrate significant improvements in performance and especially robustness across all tasks and test cases. The average performance gain on four datasets in the standard test, test with modality noise and test with missing modality over the baseline are 0.9%, 2.5%, and 2.1%.

The paper is organized as follows. Sec. 2 gives an overview of related research. Sec. 3 introduces the proposed method. Sec. 4 shows the experiment results and the analysis. Finally, Sec. 5 summarizes the paper, its limitations, and the future work.

2. Related Work

Modality alignment in multimodal learning. Almost all large-scale multimodal models use specific strategies for modal alignment. Contrastive loss is a popular approach that promotes related modality alignment by boosting the similarity of positive sample pairs (Li et al., 2021; Radford et al., 2021; Girdhar et al., 2023). Flamingo achieves modality alignment by combining a pretrained vision encoder and a language model through a series of gated cross-attention layers, allowing for effective interaction between visual and textual inputs (Alayrac et al., 2022). In (Li et al., 2023), BLIP-2 employs a lightweight Querying Transformer to connect frozen image encoders with large language models for modality alignment. In contrast, the multimodal transformer, i.e., ViLT (Kim et al., 2021) and SimVLM(Wang et al., 2021) only employ the agent task such as image text matching and word patch alignment for aligning modality during pre-training, lacking such approach at fine-tuning stage. To solve this problem, we propose an implicit regularization method to adjust the alignment of the multimodal transformer during the fine-tuning process.

Robustness of multimodal learning. Modality noise and absence are two challenges to the robustness of multimodal

learning. In ML for healthcare, the patient may be missing the data such as an X-ray due to economic/timing issues. In addition, some sensor data may be accompanied by a lot of noise due to improper wear. A similar situation occurs in vision-language tasks. For example, some online recommender models are unable to receive images uploaded by users or receive blurry images with a lot of noise due to network issues. Sijie et al. (Mai et al., 2022) propose the multimodal information bottleneck to filter out noisy information in unimodal representations. Md et al. (Islam & Iqbal, 2022) apply a cooperative multitask learning-based guided multimodal fusion approach to get robust performance on noisy and misaligned sensor data. For the missing modality problem, (Ma et al., 2021) reconstructs the missing modalities using modality priors and Bayesian Meta-Learning during the inference phase. (Lee et al., 2023) propose the missing-aware prompts to learn the patterns of complete and incomplete samples. In (Zhi et al., 2024), an approach inspired by in-context learning is proposed to improve the data efficiency for multimodal learning under missing modality and data scarcity. However, these methods require additional parameters to enhance the incomplete samples. Differently, we employ the nonparametric regularization approach to obtain robust multimodal representation.

3. Proposed Method

We first describe the problem definition and the proposed method is elaborated on later.

3.1. Problem setting

We consider the multimodal transfer learning problem with a downstream dataset $\mathcal{D}$ containing multimodal input samples. For notation simplicity, we assume there are two modalities in the dataset, i.e., $\mathcal{D} = \{x_i^{m_1}, x_i^{m_2}, y_i\}_{i=1}^N$ where y_i is the label. Note that our framework can handle any number of modalities in principle and We will describe how to extend it to 3-modalities tasks later. When we fine-tune a pretrained multimodal transformer, i.e., ViLT for solving the target task, some embedding operations are performed firstly performed on the input data $x_i^{m_1}$ and $x_i^{m_2}$:

$$x_i^{\bar{m}_1} = e_{\theta_{m_1}}(x_i^{m_1}) = [m_{1cls}; m_{11}; \cdots; m_{1L_{m_1}}], \quad (1)$$

$$x_i^{\bar{m}_2} = e_{\theta_{m_2}}(x_i^{m_2}) = [m_{2cls}; m_{21}; \cdots; m_{2L_{m_2}}], \quad (2)$$

where $e_{\theta_{m_1}}$ and $e_{\theta_{m_2}}$ refer to the embedding operation for two modalities such as linear projection, position embedding and modality type embedding (Kim et al., 2021). m_{1cls} and m_{2cls} are the added classification head token and L_{m_1} and L_{m_2} are the number of embedded tokens. $[;]$ means the concatenate operation. Then, the multimodal transformer f_{θ_X} inference the output tokens H_i by

$$H_i = f_{\theta_X}([x_i^{\bar{m}_1}; x_i^{\bar{m}_2}]) = [{H_i}^{cls}; {H_i}^{m_1}; {H_i}^{m_2}], \quad (3)$$

where ${H_i}^{m_1} \in \mathbb{R}^{L_{m_1} \times d}$ and ${H_i}^{m_2} \in \mathbb{R}^{L_{m_2} \times d}$ are the processed features for two modalities, d is the embedding dimension. ${H_i}^{cls} \in \mathbb{R}^{1 \times d}$ is the final classification head token which can be input into an added classifier/regressor f_{θ_c} for predicting the label and minimizing the loss:

$$\hat{y}_i = f_{\theta_c}({H_i}^{cls}), \ell_{task}^{(i)} = \ell_{cls}(\hat{y}_i, y_i), \quad (4)$$

where ℓ_{cls} is the task-dependent loss function such as cross-entropy and $\ell_{task}^{(i)}$ is the loss value for the ith sample.

The issue with the fine-tuning process described above is that it lacks the approach for aligning $x_i^{m_1}$ and $x_i^{m_2}$ or their representation in this multimodal transformer. We will introduce our proposed method for solving this problem in the next section.

3.2. Improve the robustness of the multimodal transformer by Wasserstein Modality Alignment

We propose the Wasserstein Modality Alignment (WMA), an implicit regularization for adjusting the alignment of different modalities in the multimodal transformer. For keeping computation efficient, we use the OT distance as the instantiation of Wasserstein distance and represent the alignment degree of two modalities by it. Uniquely, WMA search the task-dependent optimal alignment through two hyperparameters rather than directly minimizing the OT distance.

For the feature ${H_i}^{m_1}$ and ${H_i}^{m_2}$, the optimal transport problem is defined as

$$W(H_i^{m_1}, H_i^{m_2}) = \min_{T \in \Sigma(\sigma,\delta)} \langle \mathbf{C}, T \rangle, \quad (5)$$

where $\mathbf{C} \in \mathbb{R}^{L_{m_1} \times L_{m_2}}$ is a manually defined cost matrix, with each element c_{pq} representing the distance between the pth token of $H_i^{m_1}$ and the qth token of $H_i^{m_2}$. The optimal solution $T^\star$ is known as the optimal transport plan. The set $\Sigma(\sigma, \delta)$ is defined as:

$$\Sigma(\sigma, \delta) = \left\{ T \in \mathbb{R}_+^{L_{m_1} \times L_{m_2}} \mid T\mathbf{1}_{L_{m_2}} = \sigma, \, T^\top \mathbf{1}_{L_{m_1}} = \delta \right\}, \quad (6)$$

where σ and δ are the normalized distributions for $H_i^{m_1}$ and $H_i^{m_2}$, respectively, which are given by:

$$\sigma = \frac{1}{L_{m_1}} \mathbf{1}_{L_{m_1}}, \quad \delta = \frac{1}{L_{m_2}} \mathbf{1}_{L_{m_2}}, \quad (7)$$

where $\mathbf{1}_{L_{m_1}}$ and $\mathbf{1}_{L_{m_2}}$ are vectors of ones with lengths L_{m_1} and L_{m_2}, respectively. To keep the computation efficient, we apply the IPOT algorithm (Xie et al., 2018) to solve this problem.

The optimal transport cost D_{m1m2}^i is calculated as

$$D_{m_1,m_2}^i = \langle \mathbf{C}, T^* \rangle. \quad (8)$$

We use $D^i_{m_1,m_2}$ as the reference for the alignment degree and manually set the target value of $D^i_{m_1,m_2}$ to regularize the model parameters during the fine-tuning phase by modifying the loss function

$$\ell^{(i)}_{task} = \ell_{cls}(\hat{y}_i, y_i) + \alpha(D^i_{m_1,m_2} - \beta\frac{1}{bs}\sum_{j=1}^{bs} D^j_{m_1,m_2})^2, \tag{9}$$

where bs refers to the batch size. We use the average D_{m_1,m_2} of the first batch at model initialization (by applying the pretrained weight) as a basis and set the search range by a combination of hyperparameters α and β. By this strategy, we do not minimize $D^i_{m_1,m_2}$, instead, we search for the best alignment for the different modalities in the fine-tuning phase. Fig. 2 shows the search results for the Hateful Memes dataset under $\alpha = 1$ with different value of β. The best overall performance is achieved at $\beta = 2.0$ which demonstrates the superiority of the proposed WMA over minimizing OT values. For 3-modalities tasks $H_i = [{H_i}^{cls}; {H_i}^{m_1}; {H_i}^{m_2}; {H_i}^{m_3}]$, we can easily modify Eq. 9 to

$$\begin{aligned}\ell^{(i)}_{task} = \ell_{cls}(\hat{y}_i, y_i) + \alpha((D^i_{m_1,m_2} - \frac{\beta}{bs}\sum_{j=1}^{bs} D^j_{m_1,m_2})^2 \\ +(D^i_{m_1,m_3} - \frac{\beta}{bs}\sum_{j=1}^{bs} D^j_{m_1,m_3})^2 \\ +(D^i_{m_2,m_3} - \frac{\beta}{bs}\sum_{j=1}^{bs} D^j_{m_2,m_3})^2).\end{aligned} \tag{10}$$

4. Experiment

We first introduce the experimental settings and then present the experimental results of our methods and baselines on four datasets, demonstrating the effectiveness of our method.

4.1. Experimental Setting

Datasets. We select both two 2-modalities datasets and two 3-modalities datasets across different downstream tasks to evaluate our proposed method.

- Hateful Memes (Kiela et al., 2020). This is a binary classification task with two modalities image and text. The task is to detect the maliciousness of memes. The numbers of the samples in the training/val/testing dataset are 8500, 500 and 1000.
- MM-IMDb (Arevalo et al., 2017). This is a multi-label (25 labels) classification task with two modalities image and text. The task is to tag the film. The numbers of the samples in the training/val/testing dataset are 32278, 5411 and 16120.
- UR-FUNNY (Hasan et al., 2019). This is a binary classification task with three modalities text, video and audio. The task is to detect humor in talk. The number of samples in the training/val/testing dataset are 8074, 1034, 1058.
- MOSEI (Zadeh et al., 2018). This is a regression task with three modalities text, video and audio. The task is to recognize the degree of sentiment. The number of samples in the training/val/testing dataset are 16265, 1869, 4643.

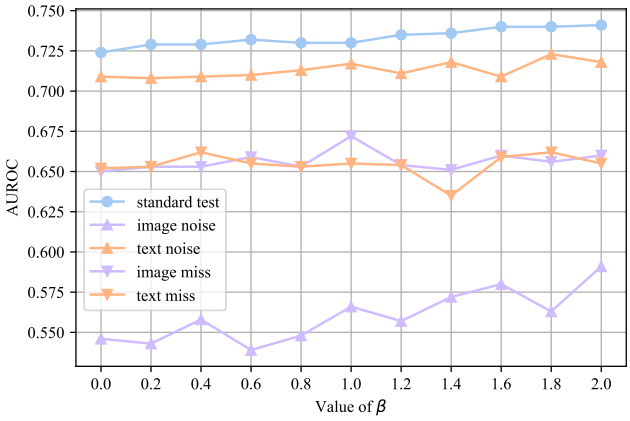

Figure 2. The performance of ViLT-WMA on Hateful Memes datasets under $\alpha = 1$ with different value of β. When $\beta = 0$, it equals to minimize the OT distance. The best overall performance is gained at $\beta = 2.0$. The performance and robustness of the model are positively correlated with the value of β, indicating that the alignment of the model should be task-dependent, instead of minimizing the OT value. See more about the relationship between performance, robustness and the value of α and β on other datasets in Table 3, 4, 5, 6 and 2.

Metrics. We set the metrics for each dataset according to the tasks. For Hateful Memes and UR-FUNNY, we use the AUROC as evaluation metrics. For MM-IMDb and MOSEI, we use F1 score and MAE, respectively.

Pretrained Multimodal Transformer. We use the classical pre-trained multimodal transformer, ViLT-B (Kim et al., 2021) as the backbone and add additional classifiers/regressors for different downstream tasks.

Input data processing. For vision-language tasks Hateful Memes and MM-IMDb, we follow the operation in ViLT (Kim et al., 2021): the text is embedded by Bert and the image is split into patches (same with ViT). For UR-FUNNY and MOSEI, we use MultiBench (Liang et al., 2021) to get the embedded feature of three modalities.

Baseline. We use the standard transfer learning approach as the baseline: adding a classifier/regressor for the target task and fine-tuning all layers of ViLT with the added layers on the target dataset.

Hyperparameter settings. We set the batch size for Hateful Memes, MM-IMDb, UR-FUNNY and MOSEI as 128, 64, 256, 256. The learning rate search range is [1e-2, 1e-3, 1e-4, 5e-5, 1e-5]. The learning rate strategy is linear decay with warm-up. The search range of α is set as [0.1, 0.2,

Table 1. Results of our proposed method with the baseline on all datasets and test cases. The bold numbers mean the best performance. The bigger AUROC and F1 and smaller MAE refer to better performance.

				test with modality noise			test with missing modality		
Datasets	Metric	Model	normal	text noise	image/video noise	audio noise	text missing	image/video missing	audio missing
Hateful Memes	AUROC	ViLT	0.736	0.711	0.571	-	0.647	0.650	-
		ViLT-WMA	**0.741**	**0.718**	**0.591**		**0.655**	**0.660**	
MM-IMDb	F1	ViLT	0.551	0.486	0.243	-	0.378	0.378	-
		ViLT-WMA	**0.558**	**0.498**	**0.255**		**0.414**	**0.384**	
UR-FUNNY	AUROC	ViLT	0.704	0.612	0.620	0.680	0.650	**0.602**	0.703
		ViLT-WMA	**0.711**	**0.634**	**0.663**	**0.694**	**0.664**	0.601	**0.707**
MOSEI	MAE	ViLT	0.807	0.814	0.817	0.807	0.806	0.824	0.823
		ViLT-WMA	**0.801**	**0.813**	**0.814**	**0.802**	**0.800**	**0.822**	**0.818**

Table 2. Results of our proposed method with minimizing OT distance on all datasets and test cases. The bold numbers mean the best performance. The bigger AUROC and F1 and smaller MAE refer to better performance.

				test with modality noise			test with missing modality		
Datasets	Metric	Regularzation	normal	text noise	image/video noise	audio noise	text missing	image/video missing	audio missing
Hateful Memes	AUROC	minimize OT	0.724	0.709	0.546	-	0.652	0.650	-
		WMA	**0.741**	**0.718**	**0.591**		**0.655**	**0.660**	
MM-IMDb	F1	minimize OT	0.550	0.487	**0.270**	-	0.379	0.381	-
		WMA	**0.558**	**0.498**	0.255		**0.414**	**0.384**	
UR-FUNNY	AUROC	minimize OT	0.707	0.564	**0.667**	0.666	0.642	0.597	0.695
		WMA	**0.711**	**0.634**	0.663	**0.694**	**0.664**	**0.601**	**0.707**
MOSEI	MAE	minimize OT	0.806	0.818	0.818	0.807	0.805	**0.821**	0.821
		WMA	**0.801**	**0.813**	**0.814**	**0.802**	**0.800**	0.822	**0.818**

1.0, 5.0]. The search range of β is [0.1, 0,2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 4.0, 6.0, 10.0]. For some of the best results obtained at the boundary values, we slightly expand the values of α and β for further searches. Early stopping with patience 5 is applied for selecting the weight. Experiments are running on the Tesla V100 GPU.

Robustness test setting. We test the robustness of the model in two ways: test with modality noise and test with missing modality. When simulating noise in a particular modality, we add uniform noise of amplitude 3 to the embedding feature. For simulating a missing modality, we randomly remove 50% of the samples for that modality by setting the embedding features all to 0.

4.2. Main Results

Table 1 presents the quantitative results of our proposed method ViLT-WMA and the baseline ViLT across all the datasets and test cases. From the table 1, we find that our proposed method significantly improved over baseline in almost all datasets and test scenarios. Under standard test, the ViLT-WMA enhance the score of four datasets at 0.7%, 1.3%, 1.0%, 0.7%. Under test with modality noise, the ViLT-WMA enhances the score of four datasets at 2.1%, 3.3%, 4.1%, 0.4%. For test with missing modality, the the score is increased for four datasets at 1.4%, 5.6%, 0.9%, 0.5%. These results demonstrate that our proposed WMA method can help the multimodal transformer to align different modes efficiently, which benefits both performance and robustness.

We also report all the search results under various combinations of α and β in Table 3, 4, 5, 6. From Table 3, 4, 5, 6 we can make the following summary:

- The modality alignment can be effectively adjusted by using our proposed WMA method. The obvious performance improvements can be achieved in almost half of the settings.
- Different tasks require different degrees of modality alignment. For example, Table 4 shows better performance and robustness at smaller target OT values for MM-IMDb dataset and the opposite trend is observed from Table 3 for Hateful Memes datasets. Our proposed method can achieve task-dependent optimal alignment.

4.3. Ablation study

We compare minimizing the OT distance with our proposed WMA method. We simulate this strategy by setting α to 1 and β to 0. The comparison of this method with our proposed method is shown in Table 2. Table 2 indicates that our proposed WMA outperforms this strategy in most of the test cases. We assume that different tasks require different levels of modality heterogeneity and alignment, and over-alignment could cause a loss of modality heterogeneity which might be important to the model performance and robustness.

5. Conclusion

This paper addresses a pivotal challenge in multimodal transformers: the absence of a modality alignment approach during the fine-tuning phase. We introduce a Wasserstein distance-based regularization method to adjust the modality alignment degree. The proposed method does not require training more parameters and can be easily integrated into the multimodal transformer. The experimental results

demonstrate significant improvements on performance and especially robustness on both 2-modalities and 3-modalities tasks. The average improvements on four datasets in the standard test, test with modality noise and test with missing modality are 0.9%, 2.5%, and 2.1% respectively. Meanwhile, our experimental results show that modality alignment needs to be task-dependent, rather than forced alignment, i.e., minimizing the OT distance between modalities, which provides valuable insights for related work. Our future work will focus on more theoretical analyses of our proposed method.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Arevalo, J., Solorio, T., Montes-y Gómez, M., and González, F. A. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017.

Ghahremani Boozandani, M. and Wachinger, C. Regbn: Batch normalization of multimodal data with regularization. *Advances in Neural Information Processing Systems*, 36, 2024.

Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K. V., Joulin, A., and Misra, I. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190, 2023.

Hasan, M. K., Rahman, W., Zadeh, A., Zhong, J., Tanveer, M. I., Morency, L.-P., et al. Ur-funny: A multimodal language dataset for understanding humor. *arXiv preprint arXiv:1904.06618*, 2019.

Hayat, N., Geras, K. J., and Shamout, F. E. Medfuse: Multimodal fusion with clinical time-series data and chest x-ray images. In *Machine Learning for Healthcare Conference*, pp. 479–503. PMLR, 2022.

Islam, M. M. and Iqbal, T. Mumu: Cooperative multitask learning-based guided multimodal fusion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 1043–1051, 2022.

Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P., and Testuggine, D. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33: 2611–2624, 2020.

Kim, E.-S., Kang, W. Y., On, K.-W., Heo, Y.-J., and Zhang, B.-T. Hypergraph attention networks for multimodal learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14581–14590, 2020.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pp. 5583–5594. PMLR, 2021.

Lee, Y.-L., Tsai, Y.-H., Chiu, W.-C., and Lee, C.-Y. Multimodal prompting with missing modalities for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14943–14952, 2023.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705, 2021.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Liang, P. P., Lyu, Y., Fan, X., Wu, Z., Cheng, Y., Wu, J., Chen, L. Y., Wu, P., Lee, M. A., Zhu, Y., et al. Multibench: Multiscale benchmarks for multimodal representation learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Liang, P. P., Cheng, Y., Fan, X., Ling, C. K., Nie, S., Chen, R., Deng, Z., Allen, N., Auerbach, R., Mahmood, F., et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Advances in Neural Information Processing Systems*, 36, 2024.

Ma, M., Ren, J., Zhao, L., Tulyakov, S., Wu, C., and Peng, X. Smil: Multimodal learning with severely missing modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2302–2310, 2021.

Mai, S., Zeng, Y., and Hu, H. Multimodal information bottleneck: Learning minimal sufficient unimodal and multimodal representations. *IEEE Transactions on Multimedia*, 2022.

Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., and Sun, C. Attention bottlenecks for multimodal fusion. *Advances in neural information processing systems*, 34: 14200–14213, 2021.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing exact wasserstein distance. *arXiv preprint arXiv:1802.04307*, 2018.

Zadeh, A., Liang, P. P., Poria, S., Vij, P., Cambria, E., and Morency, L.-P. Multi-attention recurrent network for human communication comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Zhi, Z., Liu, Z., Elbadawi, M., Daneshmend, A., Orlu, M., Basit, A., Demosthenous, A., and Rodrigues, M. Borrowing treasures from neighbors: In-context learning for multimodal learning with missing modalities and data scarcity. *arXiv preprint arXiv:2403.09428*, 2024.

A. More experimental results

We report all the search results under various combination of α and β for all datasets in Table 3, 4, 5 and 6.

Table 3. Results of different α and β on Hateful Memes dataset. Bold indicates better or equal performance than baseline, and red font is the weight we select.

	test case	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$	$\beta = 1.2$	$\beta = 1.4$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 10$
$\alpha = 5$	normal	0.721	0.727	0.733	0.732	0.734	**0.738**	0.733	**0.735**	0.721	**0.741**	**0.737**	0.731	0.714	0.728
	image noise	0.538	0.548	0.547	0.545	0.550	0.566	0.565	0.556	0.551	0.540	0.565	0.592	0.543	0.549
	text noise	0.708	0.710	0.706	0.705	0.713	**0.714**	**0.717**	**0.716**	0.708	**0.723**	**0.718**	0.712	0.701	0.698
	image missing	0.626	0.641	**0.652**	**0.659**	**0.651**	**0.663**	0.644	0.643	0.634	0.652	0.654	0.635	0.632	0.647
	text missing	0.638	**0.660**	**0.656**	0.664	**0.672**	**0.657**	**0.671**	0.641	0.633	**0.655**	**0.655**	**0.674**	0.641	**0.650**
$\alpha = 1.0$	normal	0.730	0.729	0.729	0.732	0.730	0.730	**0.735**	**0.736**	**0.740**	**0.740**	0.741	0.725	0.731	0.730
	image noise	0.540	0.543	0.558	0.539	0.548	0.566	0.557	**0.572**	**0.580**	0.563	0.591	0.548	**0.593**	**0.574**
	text noise	**0.711**	0.708	0.709	0.710	**0.713**	**0.717**	**0.711**	**0.718**	0.709	**0.723**	0.718	0.708	0.709	0.706
	image missing	**0.657**	**0.653**	**0.653**	**0.659**	**0.653**	**0.672**	**0.654**	**0.651**	**0.660**	**0.656**	0.660	0.633	**0.655**	0.646
	text missing	**0.655**	**0.653**	**0.662**	**0.655**	**0.653**	**0.655**	**0.654**	0.635	**0.659**	**0.662**	0.655	0.644	**0.649**	**0.652**
$\alpha = 0.2$	normal	0.731	0.732	0.731	**0.738**	**0.735**	**0.736**	0.732	**0.736**	**0.740**	**0.735**	0.728	0.717	**0.740**	0.721
	image noise	0.545	0.540	0.537	0.544	0.568	0.548	0.549	**0.573**	0.552	0.563	0.547	0.552	0.548	0.559
	text noise	0.710	0.708	0.710	**0.713**	**0.720**	**0.711**	**0.718**	**0.717**	**0.714**	**0.721**	0.706	0.704	**0.723**	0.705
	image missing	**0.655**	0.648	**0.652**	**0.656**	0.648	**0.658**	**0.660**	**0.651**	**0.659**	**0.656**	0.645	0.641	**0.655**	**0.656**
	text missing	**0.661**	**0.653**	**0.655**	**0.661**	**0.654**	**0.651**	**0.654**	**0.649**	**0.651**	0.643	**0.650**	0.635	**0.654**	0.642
$\alpha = 0.1$	normal	0.728	0.730	0.735	0.726	**0.739**	0.730	0.728	0.725	0.724	0.733	0.721	0.732	0.728	0.714
	image noise	0.539	0.551	0.554	0.563	0.548	0.546	0.538	0.570	0.550	0.555	0.567	**0.579**	0.543	**0.575**
	text noise	0.705	0.708	0.707	0.703	**0.714**	**0.711**	0.707	0.706	0.704	0.710	0.701	**0.719**	**0.717**	0.695
	image missing	**0.652**	**0.655**	**0.660**	0.645	**0.656**	**0.661**	0.645	0.643	0.643	0.649	0.640	0.646	**0.651**	0.639
	text missing	**0.647**	**0.652**	**0.659**	0.643	**0.658**	0.639	0.645	**0.655**	0.645	**0.651**	**0.648**	**0.658**	**0.652**	0.639

Table 4. Results of different α and β on MM-IMDb dataset. Bold indicates better or equal performance than baseline, and red font is the weight we select.

	test case	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$	$\beta = 1.2$	$\beta = 1.4$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 10$
$\alpha = 8$	normal	0.55	0.55	0.544	-	-	-	-	-	-	-	-	-	-	-
	image noise	**0.277**	**0.274**	0.24	-	-	-	-	-	-	-	-	-	-	-
	text noise	0.475	0.477	**0.497**	-	-	-	-	-	-	-	-	-	-	-
	image missing	**0.383**	**0.384**	0.366	-	-	-	-	-	-	-	-	-	-	-
	text missing	**0.409**	**0.407**	0.391	-	-	-	-	-	-	-	-	-	-	-
a=6	normal	0.55	**0.558**	**0.552**	-	-	-	-	-	-	-	-	-	-	-
	image noise	**0.283**	**0.255**	**0.271**	-	-	-	-	-	-	-	-	-	-	-
	text noise	0.473	0.498	0.485	-	-	-	-	-	-	-	-	-	-	-
	image missing	**0.384**	0.384	**0.382**	-	-	-	-	-	-	-	-	-	-	-
	text missing	**0.410**	0.414	**0.385**	-	-	-	-	-	-	-	-	-	-	-
$\alpha = 5$	normal	**0.553**	**0.555**	**0.551**	0.55	0.543	0.55	0.546	0.541	0.545	0.547	**0.551**	0.55	0.538	**0.555**
	image noise	**0.284**	**0.283**	**0.256**	0.231	**0.252**	**0.275**	**0.264**	**0.277**	**0.278**	**0.277**	**0.282**	**0.261**	0.234	0.265
	text noise	0.48	**0.496**	0.479	**0.494**	0.467	0.481	0.475	0.481	0.477	0.479	0.483	**0.495**	0.474	**0.502**
	image missing	**0.388**	**0.39**	**0.378**	0.37	0.368	0.372	0.368	0.367	0.363	0.363	0.371	0.364	0.369	**0.378**
	text missing	**0.408**	**0.396**	0.374	**0.385**	**0.405**	**0.401**	**0.39**	**0.396**	**0.394**	0.377	**0.387**	**0.384**	0.351	0.362
$\alpha = 1.0$	normal	**0.552**	**0.551**	**0.553**	0.549	0.549	0.546	0.549	0.535	0.548	0.542	0.548	0.55	0.549	**0.551**
	image noise	**0.245**	**0.255**	**0.255**	**0.254**	**0.256**	**0.267**	**0.253**	**0.254**	0.242	**0.276**	**0.276**	**0.271**	**0.249**	**0.262**
	text noise	**0.487**	**0.487**	**0.486**	0.483	0.475	0.483	0.478	0.457	0.481	**0.494**	0.484	**0.496**	0.482	0.484
	image missing	0.376	**0.381**	0.376	0.376	**0.378**	0.367	0.373	0.356	0.365	0.356	0.367	0.365	0.369	0.371
	text missing	**0.384**	**0.404**	**0.379**	**0.394**	**0.379**	**0.394**	0.37	0.376	**0.416**	**0.387**	0.372	**0.38**	**0.378**	**0.379**
$\alpha = 0.2$	normal	0.55	0.549	0.55	0.55	0.548	**0.554**	0.549	0.55	0.544	0.541	0.549	0.545	**0.554**	0.55
	image noise	**0.248**	**0.245**	**0.267**	**0.252**	**0.269**	0.241	**0.261**	**0.245**	0.239	**0.253**	0.24	**0.26**	0.242	**0.285**
	text noise	**0.487**	0.484	0.478	0.485	0.479	**0.489**	0.482	**0.488**	0.478	0.476	0.485	**0.494**	**0.498**	**0.489**
	image missing	0.377	**0.379**	**0.378**	0.372	0.376	0.374	0.37	0.371	0.364	0.363	0.368	0.359	0.368	0.37
	text missing	0.369	**0.391**	**0.39**	0.369	**0.378**	**0.421**	0.377	**0.403**	**0.396**	0.357	**0.398**	0.361	**0.394**	**0.381**
$\alpha = 0.1$	normal	**0.551**	**0.552**	**0.556**	**0.552**	**0.553**	0.55	0.55	0.549	**0.551**	**0.551**	0.549	0.547	0.549	0.546
	image noise	**0.263**	**0.258**	**0.257**	**0.266**	**0.269**	**0.286**	0.241	**0.259**	0.236	**0.249**	**0.244**	**0.272**	**0.262**	0.235
	text noise	**0.49**	0.477	**0.488**	**0.488**	**0.486**	0.477	0.482	0.478	0.484	0.485	0.476	**0.495**	**0.498**	0.473
	image missing	0.379	0.376	**0.38**	0.374	**0.379**	**0.38**	0.375	0.377	0.37	0.371	0.369	0.365	0.366	0.367
	text missing	**0.396**	0.374	**0.383**	**0.404**	0.376	**0.381**	**0.393**	0.370	**0.389**	**0.392**	**0.400**	0.374	**0.395**	0.356

Table 5. Results of different α and β on UR-FUNNY dataset. Bold indicates better or equal performance than baseline, and red font is the weight we select.

		$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$	$\beta = 1.2$	$\beta = 1.4$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 10$
$\alpha = 8$	normal	**0.707**	0.702	0.696	0.703	0.694	0.702	**0.708**	**0.706**	**0.707**	**0.706**	**0.71**	**0.712**	**0.711**	0.699
	video noise	0.591	0.596	0.586	0.619	0.578	**0.646**	**0.63**	**0.641**	0.561	0.583	**0.624**	0.588	0.581	0.589
	text noise	0.582	0.582	0.589	**0.614**	0.6	0.611	**0.636**	**0.627**	0.584	0.602	**0.614**	0.581	0.569	0.577
	audio noise	**0.693**	**0.688**	**0.683**	**0.682**	0.676	**0.68**	**0.703**	**0.697**	**0.693**	**0.693**	**0.692**	0.676	0.674	0.638
	video missing	0.578	0.579	0.582	**0.605**	0.587	**0.627**	0.585	0.599	0.57	0.589	0.587	0.589	0.586	0.585
	text missing	0.641	0.64	0.636	0.642	0.647	0.658	0.634	0.661	0.613	0.63	0.656	0.628	0.611	0.616
	audio missing	**0.706**	0.702	0.696	0.7	0.692	0.699	**0.709**	**0.705**	**0.706**	**0.705**	**0.708**	**0.709**	**0.706**	0.685
$\alpha = 6$	normal	0.701	0.699	0.701	0.701	**0.705**	**0.706**	**0.706**	**0.707**	**0.711**	**0.708**	**0.707**	**0.714**	0.703	**0.705**
	video noise	0.601	0.588	0.598	**0.623**	0.591	**0.665**	**0.666**	**0.674**	**0.622**	0.607	0.604	0.601	0.554	0.596
	text noise	0.588	0.584	0.57	**0.625**	0.59	**0.636**	**0.612**	**0.617**	**0.615**	0.604	**0.615**	0.573	0.595	0.572
	audio noise	**0.683**	**0.688**	0.677	0.678	0.679	**0.693**	**0.689**	0.678	**0.702**	**0.694**	**0.686**	0.638	0.675	0.652
	video missing	0.582	0.582	0.597	**0.615**	0.592	**0.614**	**0.617**	**0.613**	0.589	0.589	0.596	0.593	0.57	0.582
	text missing	0.638	0.644	**0.65**	0.635	0.647	**0.663**	**0.661**	**0.666**	**0.65**	0.624	**0.662**	**0.664**	0.64	0.603
	audio missing	0.701	0.698	0.7	0.697	**0.703**	**0.703**	**0.703**	**0.704**	**0.709**	**0.708**	**0.705**	0.698	0.701	0.689
$\alpha = 5$	normal	0.699	0.699	0.699	0.711	0.698	**0.704**	**0.706**	**0.707**	**0.711**	**0.709**	**0.71**	**0.714**	0.703	**0.705**
	video noise	0.592	0.587	0.601	0.663	0.592	**0.64**	**0.641**	0.612	0.559	0.59	0.599	0.617	0.54	0.607
	text noise	0.585	0.589	0.596	0.634	0.592	**0.621**	0.611	**0.613**	0.611	0.602	0.606	0.583	0.563	0.563
	audio noise	**0.689**	0.676	0.677	0.694	0.675	**0.683**	**0.687**	**0.696**	**0.703**	**0.697**	**0.695**	0.641	0.674	0.653
	video missing	0.584	0.588	0.594	0.601	0.592	**0.614**	**0.609**	0.581	0.569	0.579	0.587	**0.602**	0.577	0.585
	text missing	0.641	0.635	**0.656**	0.664	0.643	**0.658**	**0.652**	**0.655**	0.646	0.634	0.634	0.642	0.598	0.599
	audio missing	0.699	0.699	0.699	0.707	0.695	0.7	**0.705**	**0.706**	**0.709**	**0.707**	**0.708**	**0.708**	0.702	0.693
$\alpha = 1.0$	normal	0.702	0.7	0.699	0.698	0.696	0.703	0.702	0.608	0.699	0.703	**0.706**	**0.708**	**0.704**	0.702
	video noise	**0.642**	**0.637**	0.619	0.604	**0.654**	**0.633**	**0.628**	0.553	0.595	0.602	**0.645**	0.53	0.593	0.515
	text noise	0.562	0.579	**0.631**	0.593	**0.614**	0.604	0.61	0.584	**0.616**	**0.631**	**0.636**	0.59	0.582	0.557
	audio noise	0.672	0.678	**0.691**	0.677	**0.691**	**0.68**	**0.681**	0.589	**0.693**	**0.694**	**0.691**	**0.682**	0.627	0.655
	video missing	0.585	0.598	0.598	0.589	**0.606**	**0.611**	**0.613**	0.537	0.581	0.568	0.583	0.593	0.593	0.577
	text missing	**0.657**	**0.659**	**0.657**	0.641	**0.656**	0.648	0.647	0.585	0.636	0.62	0.606	0.646	**0.659**	0.642
	audio missing	0.695	0.699	0.699	0.695	0.695	0.699	0.701	0.592	0.7	0.701	0.698	**0.704**	0.678	0.693
$\alpha = 0.2$	normal	0.703	0.698	0.7	0.698	0.701	0.703	0.699	0.703	**0.707**	**0.706**	0.587	**0.708**	**0.708**	0.7
	video noise	0.61	0.611	0.616	**0.62**	0.612	**0.665**	0.601	**0.64**	**0.625**	**0.64**	0.53	**0.659**	**0.659**	0.574
	text noise	**0.612**	**0.613**	**0.612**	**0.613**	0.607	0.593	0.611	0.61	**0.619**	**0.617**	0.515	0.566	0.604	0.578
	audio noise	**0.682**	**0.681**	0.677	**0.681**	**0.687**	**0.687**	**0.684**	0.678	**0.686**	**0.687**	0.53	**0.68**	**0.694**	**0.682**
	video missing	0.6	**0.602**	**0.61**	0.589	0.585	**0.613**	0.579	**0.621**	**0.606**	**0.612**	0.533	**0.625**	**0.609**	0.583
	text missing	0.647	0.63	0.619	0.643	**0.652**	**0.651**	0.647	**0.653**	**0.652**	**0.657**	0.585	0.631	**0.662**	0.626
	audio missing	0.701	0.698	0.698	0.695	0.701	0.702	0.694	0.700	**0.706**	**0.706**	0.572	0.702	**0.708**	0.690
$\alpha = 0.1$	normal	0.694	0.695	0.698	0.693	0.7	0.703	0.702	0.702	**0.704**	0.703	0.697	0.698	0.593	**0.709**
	video noise	0.602	0.592	0.587	**0.634**	0.588	**0.642**	**0.642**	0.619	**0.633**	**0.637**	**0.646**	0.615	0.513	**0.688**
	text noise	0.605	0.597	0.598	0.592	0.604	0.565	**0.628**	0.602	0.606	0.6	**0.613**	0.607	0.535	0.582
	audio noise	**0.680**	**0.682**	0.679	0.667	0.674	0.644	**0.689**	0.679	**0.689**	0.678	0.677	0.678	0.53	0.669
	video missing	0.589	0.599	0.576	**0.606**	0.581	**0.604**	0.598	0.596	0.599	**0.621**	**0.625**	**0.619**	0.536	**0.606**
	text missing	0.62	0.628	0.647	0.638	0.639	0.635	**0.651**	0.647	0.643	0.647	**0.655**	0.637	0.585	**0.672**
	audio missing	0.693	0.695	0.698	0.692	0.697	0.693	0.701	0.702	**0.703**	0.699	0.693	0.696	0.584	0.696

Table 6. Results of different α and β on MOSEI dataset. Bold indicates better or equal performance than baseline, and red font is the weight we select.

		$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$	$\beta = 1.2$	$\beta = 1.4$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 10$
$\alpha = 5$	normal	0.808	**0.8**	**0.806**	**0.807**	0.81	**0.807**	**0.805**	0.819	0.812	0.808	0.811	0.811	0.811	0.809
	video noise	0.825	**0.817**	**0.814**	**0.817**	0.819	0.819	0.821	0.824	0.822	0.822	0.823	**0.816**	**0.816**	0.831
	text noise	0.818	0.819	0.823	0.82	0.819	0.817	0.819	0.824	0.823	0.818	0.826	0.825	0.832	0.832
	audio noise	0.808	**0.8**	**0.806**	**0.807**	0.81	0.808	**0.806**	0.818	0.812	0.808	0.811	0.811	0.811	0.809
	video missing	0.809	**0.804**	**0.805**	0.807	0.81	**0.805**	**0.806**	0.82	0.813	0.812	0.81	0.81	0.812	0.817
	text missing	**0.819**	**0.815**	**0.823**	**0.823**	0.825	**0.821**	**0.822**	**0.824**	**0.824**	0.825	0.825	0.827	0.829	0.826
	audio missing	0.824	0.824	**0.822**	**0.823**	**0.823**	**0.823**	**0.821**	0.83	0.824	**0.823**	0.824	0.824	0.824	**0.823**
$\alpha = 1.0$	normal	0.816	0.809	**0.806**	0.817	**0.805**	**0.805**	0.801	**0.804**	0.814	**0.804**	**0.805**	0.81	**0.806**	0.809
	video noise	0.823	0.818	**0.817**	0.824	0.821	**0.817**	0.814	0.82	0.822	0.818	0.818	**0.816**	0.827	0.848
	text noise	0.823	0.821	0.817	0.824	0.817	0.815	0.813	0.819	0.823	0.818	0.819	0.821	0.815	0.823
	audio noise	0.816	0.809	**0.806**	0.817	**0.806**	**0.805**	0.802	**0.804**	0.814	**0.804**	**0.805**	0.809	**0.806**	0.809
	video missing	0.817	0.809	**0.806**	0.818	**0.804**	**0.803**	0.8	**0.805**	0.814	**0.804**	**0.805**	0.809	0.807	0.812
	text missing	**0.823**	0.827	**0.822**	**0.822**	**0.821**	**0.822**	0.822	**0.82**	0.827	**0.82**	**0.823**	0.827	**0.823**	**0.823**
	audio missing	0.845	0.824	**0.822**	0.83	**0.82**	**0.82**	0.818	**0.819**	0.826	**0.819**	**0.821**	**0.823**	**0.821**	0.824
$\alpha = 0.2$	normal	**0.805**	**0.803**	0.819	**0.804**	0.809	**0.804**	**0.804**	**0.802**	**0.801**	**0.804**	**0.803**	**0.805**	**0.805**	0.809
	video noise	0.82	0.819	0.824	**0.816**	0.821	**0.816**	**0.816**	**0.812**	**0.816**	**0.815**	**0.816**	**0.815**	**0.817**	**0.814**
	text noise	0.818	0.816	0.824	0.818	0.815	0.817	0.818	0.816	0.818	0.817	0.818	0.815	0.816	0.817
	audio noise	**0.805**	**0.804**	0.818	**0.805**	0.809	**0.804**	**0.804**	**0.802**	**0.802**	**0.803**	**0.803**	**0.805**	**0.805**	0.809
	video missing	**0.804**	**0.803**	0.819	**0.804**	0.81	**0.803**	**0.803**	**0.804**	**0.801**	**0.803**	**0.802**	**0.806**	**0.806**	0.808
	text missing	**0.821**	**0.821**	**0.823**	**0.82**	**0.824**	**0.821**	**0.82**	**0.821**	**0.819**	**0.821**	**0.819**	**0.822**	**0.823**	**0.824**
	audio missing	**0.819**	**0.819**	0.833	**0.82**	**0.822**	**0.819**	**0.818**	**0.818**	**0.816**	**0.819**	**0.817**	**0.821**	**0.82**	**0.822**
$\alpha = 0.1$	normal	0.819	**0.805**	**0.805**	**0.805**	**0.805**	**0.803**	**0.804**	**0.803**	**0.803**	**0.802**	**0.802**	**0.805**	0.81	0.813
	video noise	0.824	**0.814**	**0.817**	**0.817**	**0.817**	**0.817**	0.818	**0.815**	**0.814**	**0.817**	**0.814**	**0.816**	0.829	0.819
	text noise	0.824	**0.814**	0.816	0.816	0.818	0.818	0.817	0.817	0.815	0.818	0.817	0.816	0.821	0.819
	audio noise	0.818	**0.805**	**0.805**	**0.806**	**0.806**	**0.804**	**0.805**	**0.803**	**0.803**	**0.802**	**0.802**	**0.805**	0.81	0.813
	video missing	0.819	**0.803**	**0.803**	**0.804**	**0.804**	**0.804**	**0.803**	**0.802**	**0.801**	**0.804**	**0.802**	**0.804**	0.812	0.809
	text missing	**0.823**	**0.822**	**0.821**	**0.82**	**0.821**	**0.82**	**0.821**	**0.821**	**0.821**	**0.82**	**0.821**	**0.822**	**0.823**	0.826
	audio missing	0.831	**0.821**	**0.819**	**0.82**	**0.82**	**0.818**	**0.819**	**0.818**	**0.818**	**0.818**	**0.817**	**0.819**	0.825	**0.823**