Encoding Domain Insights into Multi-modal Fusion: Improved Performance at the Cost of Robustness

Anonymous Authors[1]

Abstract

Using small-scale experiments with real and synthetic tasks, we compare multi-modal fusion methods, including a proposed ‘Product Fusion’, to demonstrate how encoding task-specific priors affects performance. Our results highlight a crucial trade-off: aligning fusion design with priors boosts clean-data accuracy with limited data but significantly diminishes robustness to noisy inputs.

1. Introduction

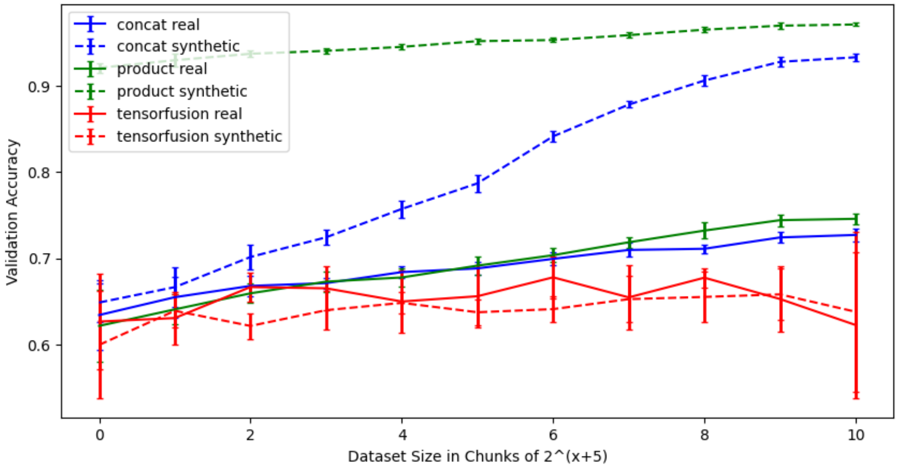

Figure 1. experimental results showing average model performance of 20 models per fusion method across randomly and progressively sampled training datasets. Error bars show standard deviation across the 20 models.

This study aims to determine: **“Under what conditions does fusion complexity impact performance concerning data availability and task complexity?”** Real-world multi-modal inputs are rarely pristine, with textual errors or low-quality visual data necessitating an understanding of model performance under varying noise conditions, an aspect often overlooked beyond raw accuracy (1). To investigate this, we employ three fusion methods: Concatenation, Tensor Fusion (2), and our proposed Product Fusion on a Twitter sentiment analysis dataset comprised of CLIP embeddings for text and images, and human-annotated sentiment labels. We aim to move towards disentangling fusion complexity from model size, and performance from mere validation accuracy.

We define “fusion complexity” along two axes: intrinsic computation (e.g., simple concatenation with no operations vs. N^2 operations in Tensor Fusion) and the resulting embedding size, which influences subsequent layer parameters (Table 1). To better isolate these factors, as intrinsic computation often correlates with embedding size, we introduce Product Fusion.

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

This method computes an element-wise product of two equal-length modality vectors (if T is text, I is image, $F_j = T_j \cdot I_j$) to match concatenation's output dimension ($2N$), where N is the dimensionality of each uni-modal input, and maintain comparable parameter counts in downstream layers, this operation is complemented by a second pass ($F_{N+j} = T_j \cdot I_{N-j}$). This design, alongside concatenation (no intrinsic computation but $2N$ output size) and Tensor Fusion (N^2 computations, $N \cdot M$ output size), allows us to test how encoding priors via intrinsic calculations impacts learning. We specifically explore this by training models on both real and simplified synthetic labels, a task for which Product and Tensor Fusion naturally compute relevant element-wise features.

"Model Performance" in this study is assessed through validation accuracy and, critically, robustness to injected Gaussian noise. This dual evaluation helps determine if designs like Product Fusion, which encode strong priors for potential accuracy gains, consequently trade off robustness. Ultimately, this experimental setup—varying fusion methods, data availability, and task nature (real vs. synthetic)—allows us to systematically investigate whether larger or more intrinsically complex models consistently achieve higher accuracy, how task alignment influences performance, and the crucial interplay between fusion design and robustness to noise.

Name	Equation	Mapping
Concatenation	$C(x_1, x_2) = (x_1, x_2)$	$C(x, y) : \mathbb{R}^{1\times n} \times \mathbb{R}^{1\times m} \rightarrow \mathbb{R}^{1\times(n+m)}$
Product Fusion	$P(x_1, x_2) = x_1 \bigotimes x_2$	$P(x, y) : \mathbb{R}^{1\times n} \times \mathbb{R}^{1\times m} \rightarrow \mathbb{R}^{1\times(n+m)}$
Tensor Fusion	$T(x_1, x_2) = x_1 \bigoplus x_2$	$T(x, y) : \mathbb{R}^{1\times n} \times \mathbb{R}^{1\times m} \rightarrow \mathbb{R}^{1\times(n\cdot m)}$

Table 1. Fusion methods. No method is parameterized, though Tensor Fusion produces a large output embedding. n,m are input vector lengths; $\bigotimes$: element-wise product; $\bigoplus$: vector outer product.

2. Related Work

Most existing work on multi-modal fusion focuses on optimizing specific fusion architectures or exploring new methods for a fixed approach, such as early or late fusion (3; 2). However, only some studies explicitly quantify how these methods behave when confronted with systematically corrupted inputs (1). Architecture search has goals similar to this work, but only in an automated fashion. Instead of explicitly comparing the performance of fusion methods, their goal is to have the model learn the fusion method itself (4), e.g., using concatenation with learned activation functions. In another recent work, the authors of (5) learn to dynamically select their fusion method from a smaller set of fusion methods. While these approaches refine fusion adaptively, they fail to directly address how well such strategies cope with noisy or partially corrupted modalities.

This work focuses on aggregation-based methods (6), which directly aggregate modalities into a latent representation rather than alignment-based methods or channel fusion techniques (6), which rely on different architectural principles (e.g., loss-based alignment or specialized shared weights for similar data types like RGB and depth).

3. Experimental Design

The model architecture comprises modality encoders (producing 768-dimensional vectors), the selected fusion method, a subsequent linear layer, and a final softmax classifier. After establishing baseline performance on clean data, we apply controlled Gaussian noise to the inputs to examine each fusion method's robustness, systematically varying the noise level to observe how performance degrades. As a synthetic task we create new labels from twitter data. Instead of the real sentiment we record the labels as the sign of the dot product between the modalities embeddings.

To explore these questions systematically, our experimentation is designed to test each task-fusion combination, evaluating them on accuracy and robustness. Simply measuring the accuracy of each model after training is sufficient to observe how model size and fusion complexity affect baseline performance. For robustness, we require more experiments to validate. Next, we apply a varying level of noise to each modality and measure accuracy under these noisy conditions. Each model predicts real and synthetic labels with random Gaussian noise sampled from $n \sim \mathcal{N}(\mu_m, \sigma_m)$ where μ_m & σ_m are the empirical mean and standard deviation of the respective modality m embeddings in the training set. A noise factor $\alpha \in [0.0, 0.95]$ scales this sampled noise, representing conditions from no to heavy corruption.. For more specificity, this

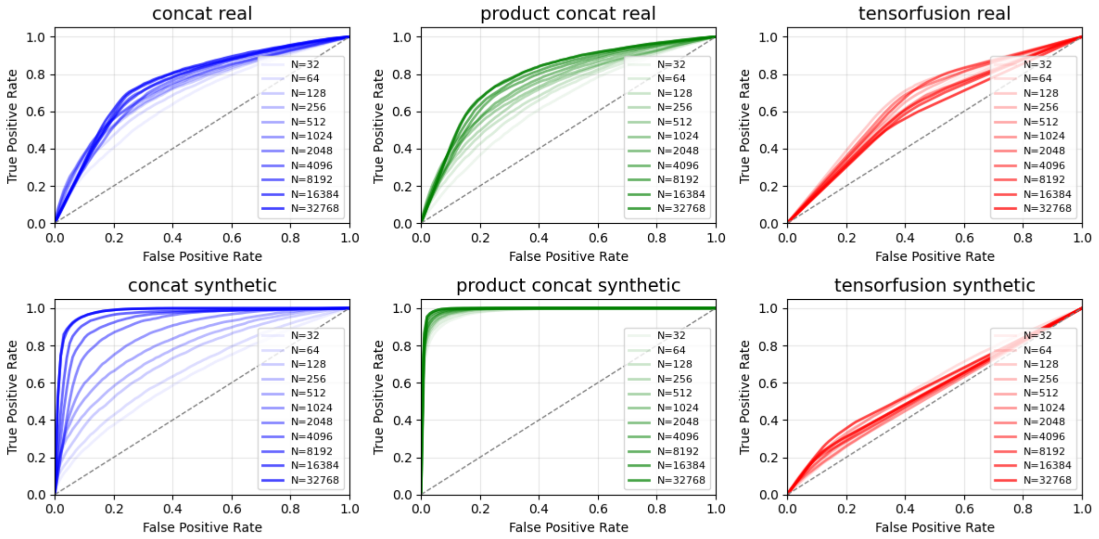

Figure 2. Receiver Operating Characteristics (ROC) curves for the 6 unit groups. Darker lines indicate larger training subsets, and color indicates the fusion method between concatenation (blue), product fusion (green), and tensor fusion (red). The figure shows significantly improved synthetic vs. real data performance, particularly for product fusion. Tensor fusion, however, fails in almost all cases.

noise is applied to each modality separately and then simultaneously for three total trials.

4. Experimental Results

4.1. Fusion-Task Alignment

We first examine performance under clean conditions and then analyze how well each model handles artificial noise. Models were trained with progressively doubled dataset sizes (32 to 327,700), averaging over 20 random initializations and data shuffles per setting. Once we train the sets of models, their final validation accuracy score is recorded as the highest validation accuracy epoch.

We then analyze how the average accuracy for each method evolves with training set size and task, focusing on their relative performance. Figures 1 and 2 show the impact of aligning the fusion method with the task. For example, Concatenation and Product Fusion have nearly identical performances on real complex data; however, when the labels are changed to be much closer to the intrinsic calculations performed for the Product Fusion method, its performance is not only significantly better but, more importantly, the data required for high performance is negligible. This clear example of the synthetic task demonstrates the effect of encoding a prior belief about the nature of the multi-modal fusion into the fusion method itself. These results establish the baseline behavior, which we will compare against noisy conditions to understand robustness.

4.2. Robustness to Noise

For robustness evaluations, we used the checkpoint of each fusion method that achieved the highest validation accuracy on clean data. The scale for noise is determined by measuring the mean and standard deviation for each modality in the test set. Overall results are shown in Figure 3.

These experiments reveal a nuanced pattern of how robustness relates to noisy inputs and fusion decisions. For instance, Tensor Fusion shows impressive robustness to noise, possibly due to its larger parameter count. In contrast, Product Fusion, which achieves the highest accuracy on clean data, struggles to perform under significant noise. Under textual noise, Product Fusion accuracy drops from $\approx 75\%$ to $\approx 55\%$ with only 10% noise when other methods are barely affected, if at all. This suggests that strong fusion-task alignment with the clean data distribution does not generalize well to noisy, unseen data. Concatenation shows moderate resilience, maintaining relatively stable performance under moderate noise conditions and combined modality corruption. These observations indicate that the interplay between fusion complexity, task-specific feature alignment, and the mode of data corruption affects the model's ability to handle noisy scenarios.

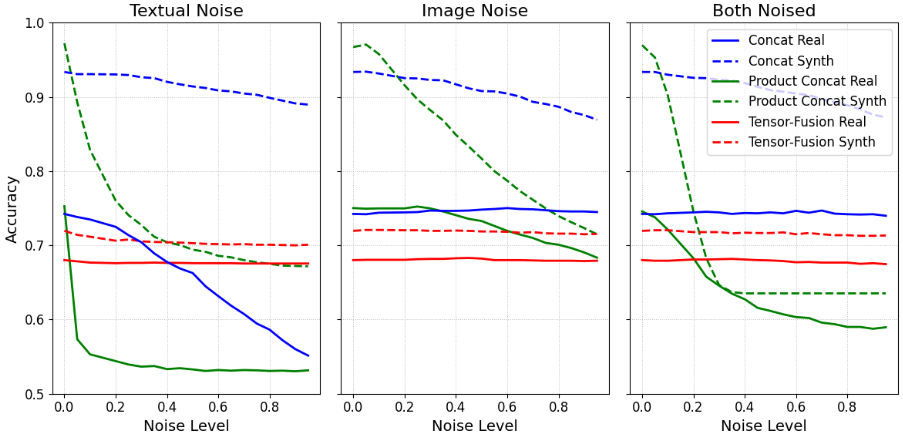

Figure 3. Model accuracy at increasing noise levels for textual (left), image (middle), and combined (right) noise conditions. Each subplot compares Product Fusion, Concatenation, and tensor-fusion methods.

5. Conclusions

Increasing model complexity does not improve model performance in this setting. Furthermore, the more parameterized models exhibit a larger variance in performance across all sizes than simpler models. Evaluating models purely on accuracy with clean data overlooks the critical dimension of robustness, as noise tests reveal stark differences in stability that accuracy alone cannot capture.

Our experiments revealed that, under these conditions, neither increased model size nor greater intrinsic complexity consistently translated to higher accuracy on the more difficult real-world task; the largest model, Tensor Fusion, generally underperformed simpler methods. While Product Fusion, with more intrinsic computations than concatenation, showed some advantage on real data with larger training sets, its primary benefit was observed on the synthetic task. Here, Product Fusion's design, which encoded priors aligned with the task (element-wise products for dot-product based labels), achieved significantly higher accuracy with minimal data. This highlights that accuracy gains from fusion complexity are critically dependent on the nature of the task and the alignment of the fusion method's implicit calculations with that task, rather than solely on model size or general complexity.

Interestingly, robustness to noise presented a contrasting trend. The largest model, Tensor Fusion, demonstrated significantly greater resilience to random noise across modalities, maintaining its relative performance even under heavy corruption. Conversely, Product Fusion exhibited substantially lower robustness despite its superior accuracy on the clean synthetic task due to strong task alignment. This suggests a critical trade-off: designs optimized for a specific clean data distribution via strong priors may not generalize well to noisy conditions, as the more straightforward learned task might result in a less resilient classifier. We also noted a more substantial reliance on textual features, though concatenation uniquely maintained performance when both modalities were noised, hinting at a potential learned mechanism for uncertain inference.

6. Limitations and Future Works

This study highlights the nuanced relationship between fusion complexity, dataset size, and task alignment, revealing the critical role of task-specific design in multi-modal learning. While our results underscore the importance of aligning fusion methods with task structure, the specific design of our synthetic task, for example, might have amplified Product Fusion's observed strengths, warranting consideration when generalizing. We were able to design a fusion method that matched our task due to its simplicity, in real world settings fusion method design will be more difficult. Future work could explore intermediate fusion complexities, the impact of data scaling, or explicit noise-robust training procedures to develop architectures that better balance high accuracy with resilience without requiring larger or more complex models.

References

[1] Konstantinos Kontras, Christos Chatzichristos, Huy Phan, Johan Suykens, and Maarten De Vos. Core-sleep: A multimodal fusion framework for time series robust to imperfect modalities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 32:840–849, 2024.

[2] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis, 2017.

[3] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[4] Juan-Manuel Perez-Rua, Valentin Vielzeuf, Stephane Pateux, Moez Baccouche, and Frederic Jurie. Mfas: Multimodal fusion architecture search. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6959–6968. IEEE, 2019.

[5] Zihui Xue and Radu Marculescu. Dynamic multimodal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2023.

[6] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4835–4845. Curran Associates, Inc., 2020.

A. Model Design and Training

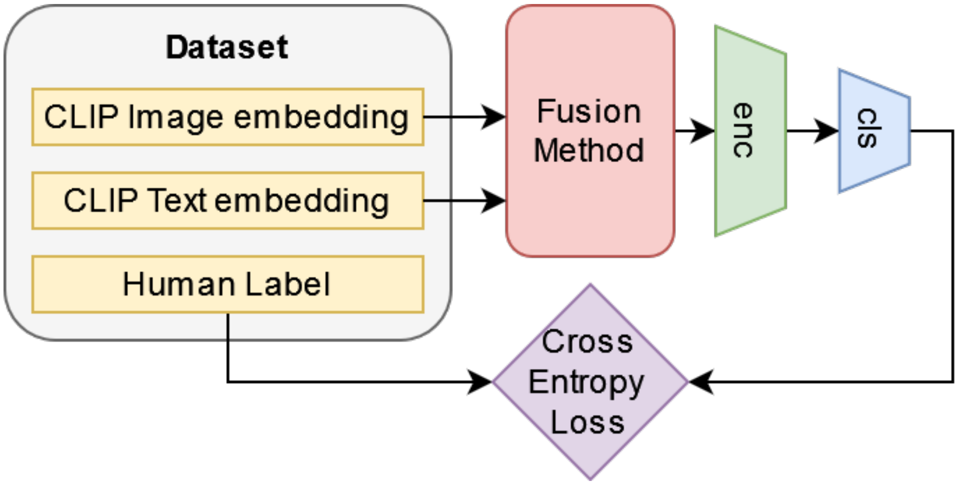

Figure 4. Model diagram of the SuperFuse model. Yellow blocks are input from the dataset, green blocks are encoders functioning as feature extractors, and blue cls block is a classifier outputting a 2-dimensional softmax classification prediction.

All fusion methods are slotted into the above model design in Figure 4. It is worth noting that while the model architectures are identical, for Tensor Fusion the resuling embedding space is significantly larger meaning the following linear layer is more parameterized than other fusion models. All models project from the fusion dimension to a 32-dimensional vector before final classification (from green block to blue in figure above)

For training all models had identical training environments. All use standard torch AdamW optimizer with a learning rate of 0.001, weight decay of 0, and 100 max epochs.