MUTUAL EQUILIBRIUM: MULTIMODAL REPRESENTATION LEARNING THROUGH RECIPROCAL FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

This work proposes a mutual feedback architecture, MEQ, that refines the two inputs, of possibly different modalities, into a pair of coupled embeddings such that each embedding reflects the information of the other. The core idea is to incorporate continuous interchange of information between the two inputs. This idea leads to a mutual feedback architecture consisting of two components whose outputs are fed back into the other. The final output of this model is defined as the fixed point of this interaction. We provide theoretical analysis that offers interpretation of this model as well as design choices to prevent failure cases. We show the benefits of MEQ through classification and visual grounding tasks spanning various datasets. Quantitatively, our model outperforms or shows competitive performance on concatenation-based multimodal classification problems. Qualitatively, the proposed interactive mechanism allows the model to progressively refine the visual grounding when paired with complementary modality, thus demonstrating the power of mutual feedback under such settings.

1 INTRODUCTION

Consider processing two inputs, $\boldsymbol{x}$ and $\boldsymbol{y}$, of possibly different modalities. The goal is to derive a representation that accounts for both inputs in the sense that the individual information are combined appropriately so as to be useful for a task involving both modalities. This problem is addressed in many disciplines such as machine learning, modality fusion, and data mining. It has become one of the most important preprocessing steps in multimodal domains.

A de facto standard way of approaching this problem is to concatenate $\boldsymbol{x}$ and $\boldsymbol{y}$ before passing them through a neural network in the hope that the information are sufficiently fused. However, existing methods typically treat the individual features as fixed descriptors and combine them in a single feed-forward operation. Such formulations implicitly assume that each extracted embedding is already an optimal representation. But complementary modalities often provide information that should alter how the other modality is interpreted. For instance, an image of a tiger hidden in bushes paired with a text reading "A tiger staying hidden" should have a representation that reflects the 'tiger-ness' more strongly than one without such text. Conversely, the corresponding text representation should also be updated to include the information that the tiger is hiding in bushes.

We therefore argue that multimodal representation learning is more naturally viewed as a process where inter-modality influence plays an important role. Specifically, we propose an iterative process in which modality-specific representations repeatedly influence one another until reaching a mutually consistent state. This idea resembles how two humans update their beliefs through repetitive exchange until their beliefs agree. It is this iterative influence that we aim to model mathematically and derive an algorithm that can refine the two inputs into another pair where each new representation will reflect information in the other.

Furthermore, we propose to model this iterative procedure as a pair of neural networks that feed each other, where one's output becomes part of the input to the other (Fig. 1). The final output of this architecture is the *fixed point* of this iteration, which is a natural way to embody the notion of agreement: when the information exchange no longer updates each other's belief, an agreement is said to be achieved. Thus, our central contribution becomes a mutually interactive system that achieves equilibrium.

Although we primarily investigate our algorithm in a representation learning framework, such an iterative refinement approach can have numerous applications in other areas. For example, cases when

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

we need to correct conflicting sensor inputs, or improve LLMs through iteration are our intended future works.

1.1 Related Works and Contributions

In this section, we provide a list of works relevant to ours, focusing on the following three areas that are related to our topic.

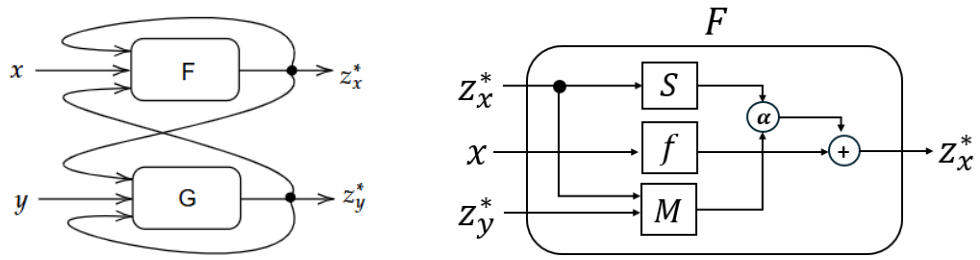
Multimodal fusion Although we do not deal with multimodal fusion directly, we review this field as it somewhat overlaps with ours in application domain. Multimodal fusion has long been considered the essential step in multimodal learning. While there are numerous prior work dealing with multimodal fusion itself, a good majority of them attempt variations of concatenation of features (Baltrušaitis et al., 2019; Manzoor et al., 2024). However, there are a few that share the spirit of our approach. The tensor fusion network (TFN: Zadeh et al. (2017)) is one of the earlier works that address fusion by means of intra- and inter-modality connection instead of relying on simple concatenation. Co-attention network (Yu et al., 2019) and LXMERT (Tan & Bansal, 2019) follow a similar path, except they specifically use cross-modal attention layers in place of TFN's outer products, to train text and vision encoders via cross-attention. Also, TFN's high computational complexity was later addressed by low-rank multimodal fusion (LMF: Liu et al. (2018)), while multimodal compact bilinear pooling (MCB: Fukui et al. (2016)) had earlier reduced the cost of bilinear interaction by sketching the outer product. MulT (Tsai et al., 2019) extended the idea to stacks of multimodal cross-attentions in a Transformer architecture to achieve repeated interaction between modalities. However, MulT limits the number of interactions to a fixed finite number, whereas we approach it with a fixed point.

Modality interaction The idea of inputs interacting with each other has also been used in other settings. For example, self-supervised approaches like BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) demonstrate the value of repeated information exchange. However, such approaches also perform finite number of interactions. In addition, such a concept has also been extended to mixture of experts (Xin et al., 2025) and graph node classification (Li et al., 2025).
Several more recent works have expanded the fusion domain to that of large language models (LLMs). In particular, it is worth noting that the concept of *multi-agent LLMs* was proposed in the context of repeated information exchange. These works are based on the idea that multiple LLMs can refine the output by interacting with each other's outputs (Wang et al., 2025; Du et al., 2024). The crucial difference between these works and ours is that first, we operate in representation space, widening the applicability, and secondly we theoretically analyze the dynamic coupling of the two encoders.

Deep equilibrium models Heavily based on the concept of a fixed point, our model naturally falls into the deep equilibrium model (DEQ: Bai et al. (2019)) family. These are models that return the fixed point of iterated hidden state updates as outputs. We will borrow common tools such as proofs and Jacobian analysis (Bai et al., 2021; 2020) from such works when analyzing our algorithm. Of particular interest among these models is the work on deep equilibrium for multimodal fusion (Ni et al., 2023). While this work does share a theme with our work, the main approach relies on a simple DEQ application to weighted sum of features (i.e., a single hidden state). Again, our work focuses on how the coupled dynamics arise as the two modalities interact, and we compare against that fusion design directly in Section 3.3.

That said, our contributions can be summarized as follows.

- We formulate mutually interactive feature learning as a coupled dynamic system, whose final outcome is the fixed point of the interaction. The interaction is performed in representation space rather than in output space, allowing for further applications.
- Theoretical properties of the said system are identified and used in deriving the main algorithm. This also results in a framework on which variant approaches can be based by allowing for different modules that fit the theoretical profile.
- Our experiments verify that such an equilibrium is meaningful in a variety of prediction tasks. Moreover, we show there is value in iterative refinement itself.



(a) Overall diagram (b) Template for $F(x, \boldsymbol{z}_x^*, \boldsymbol{z}_y^*)$.

Figure 1: Diagrams for the main architecture. Left is the overall architecture, and the right is the schematic diagram for F (G is symmetric).

2 Main Approach

Given two inputs (or modalities) $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ and $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^m$, the goal is to generate a pair of embeddings $\boldsymbol{z}_x \in \mathbb{R}^k$ and $\boldsymbol{z}_y \in \mathbb{R}^k$ such that they are mutual reflections of the other inputs[1].
The high-level idea is to first derive an intermediary representation by reflecting $\boldsymbol{y}$'s information to $\boldsymbol{x}$. Then this intermediary information should also be used to collect $\boldsymbol{x}$'s information into $\boldsymbol{y}$. Ideally, such a back-and-forth should run infinitely long in order to properly reflect the mirrored information in both inputs. We define the stopping point of this refinement to be the fixed point of this iteration.

To formally describe this setting, let F_θ and G_ϕ be the two feature extractors for the two inputs $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively parameterized by sets of parameters θ and ϕ. Each feature extractor will produce a hidden state that summarizes the information contained in the two inputs. More precisely, $F_\theta : \mathcal{X} \times \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$ is a function that takes $\boldsymbol{x}$, its own hidden state, and the other hidden state as input, and produces an updated hidden state $\boldsymbol{z}_x \in \mathbb{R}^d$. This other given hidden state is meant to represent the summary of $\boldsymbol{y}$ given $\boldsymbol{x}$. Conversely, $G_\phi : \mathcal{Y} \times \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$ achieves the same effect on $\boldsymbol{y}$, providing the necessary input $\boldsymbol{z}_y$ to F_θ (Fig. 1a).

Eventually, our proposed model will take a mutually recursive form where the output of one component is fed back into the other, and vice versa. When viewed as an iterative process, one could define a single step of update as follows (left):

$$\begin{cases} \boldsymbol{z}_x^{(k+1)} \leftarrow F_\theta(\boldsymbol{x}, \boldsymbol{z}_x^{(k)}, \boldsymbol{z}_y^{(k)}) \\ \boldsymbol{z}_y^{(k+1)} \leftarrow G_\phi(\boldsymbol{y}, \boldsymbol{z}_y^{(k)}, \boldsymbol{z}_x^{(k)}) \end{cases} \quad \underset{k \to \infty}{\Rightarrow} \quad \begin{cases} \boldsymbol{z}_x^* = F_\theta(\boldsymbol{x}, \boldsymbol{z}_x^*, \boldsymbol{z}_y^*) \\ \boldsymbol{z}_y^* = G_\phi(\boldsymbol{y}, \boldsymbol{z}_y^*, \boldsymbol{z}_x^*) \end{cases} \tag{1}$$

However, as the right-hand side of the above equation states, the ultimate outcome are the fixed points of such iterations. We compactly represent this in terms of the joint operator $T \triangleq [F_\theta; G_\phi]$:

$$\boldsymbol{z}^* = T(\boldsymbol{u}, \boldsymbol{z}^*), \quad \text{where } \boldsymbol{u} \triangleq [\boldsymbol{x}; \boldsymbol{y}] \text{ and } \boldsymbol{z}^* \triangleq [\boldsymbol{z}_x^*; \boldsymbol{z}_y^*]. \tag{2}$$

The Jacobian J_T^* of T at $\boldsymbol{z}^*$ takes a block form as follows.

$$\frac{\partial J}{\partial \boldsymbol{z}^*} \triangleq J_T^* = \begin{pmatrix} \partial F/\partial \boldsymbol{z}_x^* & \partial F/\partial \boldsymbol{z}_y^* \\ \partial G/\partial \boldsymbol{z}_x^* & \partial G/\partial \boldsymbol{z}_y^* \end{pmatrix} \triangleq \begin{pmatrix} F_{\boldsymbol{z}_x^*} & F_{\boldsymbol{z}_y^*} \\ G_{\boldsymbol{z}_x^*} & G_{\boldsymbol{z}_y^*} \end{pmatrix}. \tag{3}$$

While the right-hand side of Equation 1 suggests that a *fixed point iteration* (FPI) can be used to find such fixed points, it is common to rely on solver-based fixed point computation such as Anderson acceleration (Anderson, 1965) or Broyden's algorithm (Broyden, 1965). In practice we use a damped FPI truncated at a fixed number of steps K, and read the representation at $\boldsymbol{z}^{(K)}$ rather than at a root. The reason for this is due to the nature of our approach: Being a coupled dynamic system, using solver-based approaches in our setting can fail numerically due to sensitivity with respect to hyperparameters. As such, we rely on a more robust FPI, combined with a residual penalty term described in Sec. 2.2 (also see Tbl. 9).

Notice that Equation 2 is precisely the formulation given in DEQ (Bai et al., 2019). Because our model is a special case of DEQ, it also inherits the theoretical guarantees, such as gradient

[1]We discuss the case of more than two modalities in Appendix B

computation and universality. However, ours differs from DEQ in how the equilibrium arises: DEQ is essentially a single-step computation, whereas our model computes the fixed points by means of *interaction* between the two inputs. The presence of such interactions requires us to carefully design the underlying architecture (Sec. 2.1).

Another key characteristic of our model is that it produces a pair of *coupled* representations. That is, the $\boldsymbol{z}_x$ and $\boldsymbol{z}_y$ do not stand independently but rather jointly, each containing information from the other modality. We hypothesize that such paired embeddings will allow us to better represent the complementary information present in both inputs by iteratively referring to and updating each other's beliefs.

2.1 Theoretical Properties and Design Choices

In this section, we lay out theory-inspired design details for F_θ and G_ϕ. We first state a simple lemma that will assist our analysis.

Lemma 1. *Given Equation 1, the Jacobians of $\boldsymbol{z}_x^*$ and $\boldsymbol{z}_y^*$ are given as*

$$\frac{d\boldsymbol{z}_x^*}{d(\cdot)} = \left(I - F_{\boldsymbol{z}_x^*} - F_{\boldsymbol{z}_y^*}\left(I - G_{\boldsymbol{z}_y^*}\right)^{-1} G_{\boldsymbol{z}_x^*}\right)^{-1} \left(\frac{\partial F}{\partial(\cdot)} + F_{\boldsymbol{z}_y^*}\left(I - G_{\boldsymbol{z}_y^*}\right)^{-1} \frac{\partial G}{\partial(\cdot)}\right)$$
$$\frac{d\boldsymbol{z}_y^*}{d(\cdot)} = \left(I - G_{\boldsymbol{z}_y^*} - G_{\boldsymbol{z}_x^*}\left(I - F_{\boldsymbol{z}_x^*}\right)^{-1} F_{\boldsymbol{z}_y^*}\right)^{-1} \left(\frac{\partial G}{\partial(\cdot)} + G_{\boldsymbol{z}_x^*}\left(I - F_{\boldsymbol{z}_x^*}\right)^{-1} \frac{\partial F}{\partial(\cdot)}\right),$$

where $(\cdot)$ is a placeholder indicating any independent variable (e.g., θ or $\boldsymbol{x}$).

Proof. Simple extension of the one given in Bai et al. (2019). See Appendix A.1. □

The Lemma gives the gradient of the equilibrium with respect to the parameters:

$$\frac{\partial \mathcal{L}}{\partial \Theta} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}_x^*}\frac{d\boldsymbol{z}_x^*}{d\Theta} + \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d\Theta}, \tag{4}$$

where $\mathcal{L} = \mathcal{L}(\mathcal{D}; \Theta)$ is the loss function involving the parameter set $\Theta = \{\theta, \phi\}$ and the dataset $\mathcal{D}$ (Sec. 2.2).

In practice gradients are obtained by backpropagating through the K unrolled updates rather than by an implicit solver, so Lemma 1 is the analytical tool used above rather than the implemented gradient path (Appendix C.1).

But Lemma 1 can also be used to analyze how and when a failure occurs in the proposed architecture, especially by means of Jacobian analysis (Geng et al., 2021; Bai et al., 2021). Viewing the fixed points as a function of the original inputs, we can interpret the lemma as stating how much raw information is reflected in the fixed point. Of particular interest is the pair of inverse terms appearing in both equations. These inverse terms are independent of the placeholder $(\cdot)$, and they capture the total amount of informational influence being passed between the two modalities. We thus call these inverses the *total influence terms*. Please see Appendix A.2 for a detailed analysis of this term.

Sensitivity Another aspect of the importance of the influence term is related to the sensitivity of the fixed points with respect to the raw inputs. That is, following the compact notation given by Eqn. 2, we look at the amount of perturbation $\epsilon_{\boldsymbol{z}}$ of the fixed point $\boldsymbol{z}^*$ when the input $\boldsymbol{u} = [\boldsymbol{x}; \boldsymbol{y}]$ gets perturbed by a small amount $\epsilon_{\boldsymbol{u}}$. Standard Taylor's expansion yields the following approximate bound (derivation in Appendix A.3):

$$\|\epsilon_{\boldsymbol{z}}\| \lesssim \left\|(I - J_T^*)^{-1}\right\| \left\|\frac{\partial T}{\partial \boldsymbol{u}}\right\| \|\epsilon_{\boldsymbol{u}}\|.$$

The inverse term on the right hand side is the gathering of the total influence terms appearing in Lemma 1. This also signifies how the overall influence affects the sensitivity of the fixed point. An important consequence of this derivation is that the smaller the spectral radius $\rho(J_T^*)$, the more stable the fixed points are. This fact suggests two possible design choices: (1) structurally constrain the operator T to have $\rho(J_T^*)$ less than 1, or (2) incorporate the spectral radius into the penalty term in the main objective.

One obvious way to ensure the first approach is to let T be contractive, or 1-Lipschitz (Serrurier et al.,

2023; Tanielian & Biau, 2021) in the hidden states[2]. However, while certain classes of 1-Lipschitz neural networks are proven to be universal function approximators (Anil et al., 2019), making a function globally contractive could be detrimental in our case (see 'Fixed point collapse' below). Hence, we opt for the second option of regularization.

Cross-modal sensitivity We can also analyze how sensitive the fixed points are with respect to the inputs $\boldsymbol{x}$ and $\boldsymbol{y}$. In particular, we analyze $d\boldsymbol{z}_x^*/d\boldsymbol{y}$ and $d\boldsymbol{z}_y^*/d\boldsymbol{x}$, which reflect how much of the raw inputs are reflected in the other embedding. By plugging in $\boldsymbol{x}$ and $\boldsymbol{y}$ into Lemma 1, we get

$$\frac{d\boldsymbol{z}_x^*}{dy} = \left(I - F_{\boldsymbol{z}_x^*} - F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1}G_{\boldsymbol{z}_x^*}\right)^{-1} F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1}\frac{\partial G}{\partial y}$$
$$\frac{d\boldsymbol{z}_y^*}{dx} = \left(I - G_{\boldsymbol{z}_y^*} - G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}F_{\boldsymbol{z}_y^*}\right)^{-1} G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}\frac{\partial F}{\partial x}. \tag{5}$$

Assuming that the inverses exist, the fact that these Jacobians are close to zero means that the fixed points computed have been detached from the other raw input. In order to prevent this failure mode, $F_{\boldsymbol{z}_y^*}$ and $\partial G/\partial y$ (resp., $G_{\boldsymbol{z}_x^*}$ and $\partial F/\partial x$) should be non-zero. This immediately suggests two conditions: (1) F and G should be using $\boldsymbol{z}_y^*$ and $\boldsymbol{z}_x^*$, respectively, non-trivially, and (2) F and G should be using x and y, respectively, non-trivially. That is, both the cross-modal hidden states and raw inputs must be used in the other functions. We thus propose to use the following templates for F and G (Fig. 1b):

$$F(x, \boldsymbol{z}_x^*, \boldsymbol{z}_y^*; \alpha) = \alpha S(\boldsymbol{z}_x^*) + (1-\alpha)M(\boldsymbol{z}_x^*, \boldsymbol{z}_y^*) + f(x)$$
$$G(y, \boldsymbol{z}_y^*, \boldsymbol{z}_x^*; \alpha) = \alpha S(\boldsymbol{z}_y^*) + (1-\alpha)M(\boldsymbol{z}_y^*, \boldsymbol{z}_x^*) + g(y), \tag{6}$$

where α is a learned mixing parameter and $f(\cdot), g(\cdot)$ are raw input feature extractors for x and y, respectively. S is the self-influence function, and M is the cross-modal influence function. These are any differentiable functions that extract uni- and bimodal information from the hidden states.
Such a design is inspired by how the total influence decomposes into sums of self- and cross-modal influence terms. Even though it has no direct theoretical consequences, having both types of influences in F and G will ensure the existence of the Jacobians in Equation 12. The f and g terms added in Equation 6 are to ensure the non-zero cross-sensitivity given by Equation 5. In our vision-language experiments S is a self-attention block and M a cross-attention block in which the other modality supplies keys and values; on CMU-MOSEI both are replaced by a per-modality MLP over its own injection, its own state and the other states.

Fixed point collapse Next, we look at a failure case of our model that we term *fixed point collapse*. This failure happens when both fixed points $\boldsymbol{z}_x^*$ and $\boldsymbol{z}_y^*$ get mapped to a single constant regardless of $\boldsymbol{x}$ and $\boldsymbol{y}$. An analytical way of seeing this is when the norms of the Jacobians $d\boldsymbol{z}_x^*/dx$ and $d\boldsymbol{z}_y^*/dy$ given by Lemma 1 become close to 0. This is because the functions F and G can be thought of as being parameterized by x and y, respectively, and the fixed points should be determined by the input values under normal circumstances. If not, the fixed points are independent of the inputs, hence fixed constants.
A clear case of this happens when the sub-Jacobians $\partial F/\partial x$ and $\partial G/\partial y$ are zero – i.e., the functions F and G do not use the raw inputs. So it is clear that the two modules must use the inputs in a non-trivial way. In conjunction with these sub-Jacobian norms being close to zero, a collapse happens if the norm of the total influence is very small at the same time. Indeed, the norm of the input sensitivity, $d\boldsymbol{z}_x^*/d\boldsymbol{x}$, is bounded as follows (Let $(\cdot) = \boldsymbol{x}$ in Lemma 1):

$$\left\|\frac{d\boldsymbol{z}_x^*}{d\boldsymbol{x}}\right\| \leq \left\|\left(I - F_{\boldsymbol{z}_x^*} - F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1}G_{\boldsymbol{z}_x^*}\right)^{-1}\right\| \left\|\frac{\partial F}{\partial \boldsymbol{x}}\right\| \tag{7}$$

So even if $\partial F/\partial \boldsymbol{x}$ is non-zero (but small), a fixed point collapse effect can still happen if the norm of the total influence is small too. Note that this partially contrasts the suggestions made in the sensitivity discussion. In order to minimize the perturbation error, it was advised that $\rho(J_T^*)$ be as small as possible. However, it also turns out that having too small a $\rho(J_T^*)$ can induce a near-fixed point collapse, albeit conditioned on the small sub-Jacobian norm. One way to view this is: If the total influence is too large, the model will be overflown with influence making the fixed points too sensitive. If it is too small, then there will not be enough momentum to produce meaningful fixed points.

[2]A *contractive map* f satisfies $\|f(x) - f(y)\| < \|x - y\|$ for all possible x and y in f's domain.

2.2 TRAINING OBJECTIVE

The main training objective of MEQ depends on the task on which it is being trained. In addition to the main objective $\mathcal{L}_{task}$, we add two regularization terms to address the theoretical issues raised.

In order to balance expressivity and collapse prevention, the spectral radius $\rho(J_T^*)$ needs to be smaller than 1, but not too small. Although this is conditioned on the $\partial J/\partial \boldsymbol{x}$ being small too, we propose to use a range penalty on $\rho(J_T^*)$ around the fixed point as a safety measure.

$$\mathcal{L}_{jac} = \left(\max\left\{0, \|\hat{J}_T\| - \rho_h\right\}\right)^2 + \left(\max\left\{0, \rho_\ell - \|\hat{J}_T\|\right\}\right)^2,$$

where ρ_h and ρ_ℓ are the hyperparameters used to put the Jacobian norm inside the range $[\rho_\ell, \rho_h]$. Empirically, this also has the effect of making the computation deeper. That is, too small a spectral radius will make the fixed points too easy to reach, which in turn loses the benefits of iterative refinement. Forcing a lower bound on the radius will result in a more expressive function.
We use the estimate $\|\hat{J}_T\|$ of the true Jacobian norm around the fixed point $\boldsymbol{z}^*$, since full computation is expensive. For example, we use the Hutchinson estimator (Hutchinson, 1990) in our experiments.

The second regularization term we propose is called the *fixed point correction* term. For that, we first define the fixed point residual given the input $\boldsymbol{u} = [\boldsymbol{x}; \boldsymbol{y}]$ as follows:

$$r(\boldsymbol{z}^*; \boldsymbol{u}) \triangleq \boldsymbol{z}^* - T(\boldsymbol{u}, \boldsymbol{z}^*)$$

Then we define the fixed point correction regularizer as the sum of

$$\mathcal{L}_{fpc} = \frac{1}{2} \sum_{m \in \{\boldsymbol{x}, \boldsymbol{y}\}} \frac{\|r(\boldsymbol{z}^*; \boldsymbol{u})_m\|}{\|\boldsymbol{z}_m^*\|},$$

where the subscript m indexes one of the two input modalities. While $r(\boldsymbol{z}^*; \boldsymbol{u})$ should be zero in theory, practical limitations or settings often make it nonzero. For instance, a fixed iteration budget K leaves a nonzero residual whenever the map is slow to settle, so the representation we read is only an approximate fixed point. High value of $\mathcal{L}_{fpc}$ indicates those difficult cases. Penalizing the model using this term will update the model (e.g., by flattening) to yield fixed points that can be reached through small number of iterations.
Empirically this term is what keeps the truncated iterate close to self-consistent(Table 9 in Appendix C.3). The final objective we propose is thus a weighted sum of these regularizers plus the main task loss.

$$\mathcal{L}(\mathcal{D}; \Theta) = \mathcal{L}_{task} + \lambda_j \mathcal{L}_{jac} + \lambda_f \mathcal{L}_{fpc} \tag{8}$$

3 EXPERIMENTS

The central goal of the experiments presented here is to demonstrate the value of mutual iterative refinement of features. That is, whether the prediction can be improved or recovered from the initial faulty guess through repeated refinement, rather than achieving state-of-the-art performance. We first describe the experimental setups chosen to verify this property. For ease of presentation, we denote our proposed model as MEQ (**M**utual **EQ**uilibrium model). Please refer to Appendix C.1 for details regarding implementation and experimental setups. Appendices C.2 and C.3 contain more experimental results and analyses.

Baselines While we acknowledge there are many multimodal fusion techniques, we rely on simpler baselines in order to eliminate the extra architectural innovations and isolate the effects of the proposed structure. The baseline we compare against are simple concatenation-based fusion and attention-based fusion. The attention-based fusion approach is further divided into two variants: (1) perform self attention on the concatenated inputs $[\boldsymbol{x}; \boldsymbol{y}]$, and (2) perform cross-attention on the two inputs. In comparison, our model produces a pair of vectors $\boldsymbol{z}_x^*, \boldsymbol{z}_y^*$, which are concatenated to form the final feature. All attention models were adjusted to match the parameter count to that of our model.

Datasets We evaluate on the following benchmarks spanning different tasks and modalities: Hateful Memes (Kiela et al., 2020), VCR (Zellers et al., 2019), VQA v2 (Goyal et al., 2017), SNLI-VE (Xie et al., 2019), and CMU-MOSEI (Zadeh et al., 2018). Please refer to Appendix C.1 for detailed statistics of these datasets.

3.1 ACCURACY RESULTS

We first verify the usability of MEQ in the prediction domain. Our MEQ improves over simple concatenation on four of the five benchmarks (Table 1). The largest gain is on VQA v2 ($+14.36$ pp), and the exception is CMU-MOSEI, where the two are within seed noise of each other (-0.59 pp against a two-sigma spread of 1.10 pp). Against the stronger of the two parameter-matched attention baselines, MEQ leads on SNLI-VE, VCR and VQA v2 by smaller margins ($+0.74$ to $+0.94$ pp), and trails on Hateful Memes (-2.31 pp). Regarding the CMU-MOSEI results being indistinguishable from both baselines, further examination showed that this is due to the characteristics of the dataset itself, rather than the structural design choices (Sec. 3.3).

Table 1: Accuracy results across multimodal benchmarks. All numbers are means over three seeds under an identical training recipe. Best attn. is the stronger of parameter-matched cross- and self-attention. VQA v2 and VCR use non-standard evaluation subsets; see Appendix C.1.

Dataset	Metric	Concat	Best attn.	MEQ	Δ concat / attn.
Hateful Memes	AUROC	0.6873	0.7251	0.7021	+1.48 / -2.31pp
VCR	Acc	56.38	62.09	**62.95**	+6.57 / +0.85pp
VQA v2	Acc	48.97	62.39	**63.33**	+14.36 / +0.94pp
SNLI-VE	Acc	71.61	74.40	**75.15**	+3.53 / +0.74pp
CMU-MOSEI	Acc-7	54.79	54.74	54.20	-0.59 / -0.54pp

While MEQ outperforms simple concatenation, there seems to be little difference between the attention-based baselines. We conjecture this is because of how attention mechanisms are already powerful modality fusion algorithms (Gkoumas et al., 2021; Xu et al., 2023). Adding to that is the fact that the CLIP and BERT encoders for both modalities are state-of-the-art. This is somewhat expected, since the strength of our model is not in modality fusion but in iterated reasoning. We show this evidence in the next section. Iterating to the read depth costs $2.13\times$ the wall clock of the fastest single-pass baseline, while peak memory does not grow with K (Table 13).

3.2 ITERATIVE IMPROVEMENT RESULTS

In this section, we verify how MEQ can improve or update the performance as it performs iteration. As the first example, we visualize how the visual grounding changes along with the fixed point iteration progress. Specifically, we test our model on the COD10K (Fan et al., 2020) dataset, where the task is to locate various real-world objects that lay hidden against natural backgrounds. We first train our MEQ model on the SNLI-VE dataset. Then at inference time, we feed the raw image as the $\boldsymbol{x}$ component, and the paired $\boldsymbol{y}$ component is a text of the form *"A* [NAME] *is hidden in the picture"*, where 'NAME' is replaced with the ground truth label. Then we see which part of the image is focused at as the fixed point iteration goes on. Figure 2 shows the Grad-CAM (Selvaraju et al., 2017) visualization of the object at different iterations. On the success cases (top three rows), attention focuses on the hidden objects within the first few iterations, while in the failure case (bottom row) never localizes.[3]

In addition to the heat maps, we also show how the prediction probability changes. The accuracy vs. k plot is shown in Figure 3: The probability assigned to entailment for the matched caption (solid) and for a mismatched caption (dotted), over $k \in [1, 20]$ with the read depth $K{=}10$ marked. Dataset-level accuracy peaks near K (80.0 at $k{=}10$ vs. 77.2 at $k{=}300$).

The dotted green curves on the plots are the accuracy change when the image is paired with a completely irrelevant text (e.g., "A man is walking down the street."). Here, we can observe that the model increases (and decreases) the correct (and incorrect, resp.) probabilities for the success cases.

3.3 ABLATION STUDIES

We ablate each component on VQA v2, the benchmark where MEQ's gain over concatenation is largest, so the ablation measures the components where coupling demonstrably pays (Table 2).

[3]Results of training and testing on it directly are in Table 8 (Appendix C.2).

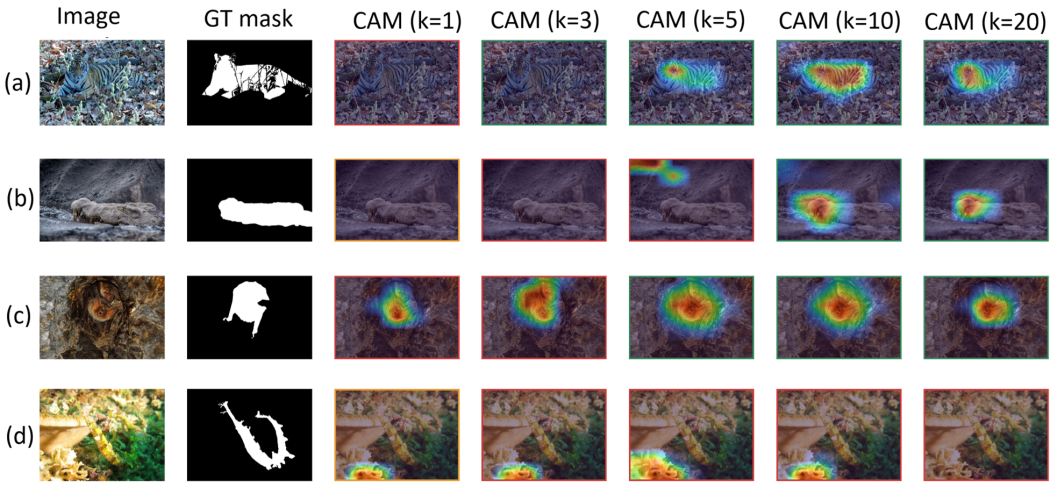
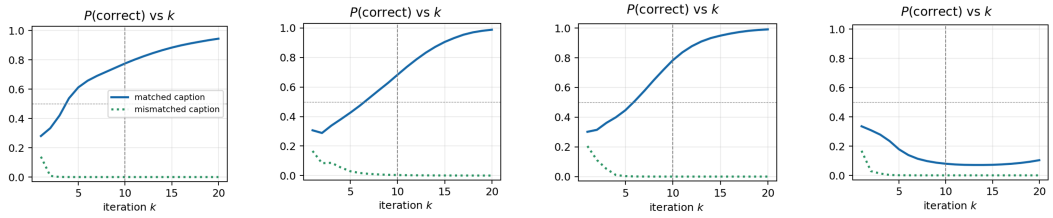


Figure 2: Application of MEQ (trained on SNLI-VE) to camouflaged-object images from COD10k, where ground-truth masks allow direct verification of visual grounding. Border colors give the prediction at each step (green: entailment, red: contradiction, orange: neutral). Best viewed in color.



(a) Tiger (success) (b) Crocodile (success) (c) Bat (success) (d) Pipefish (failure)

Figure 3: Plots of $P(\text{correct})$ vs. k.

Removing cross-attention costs 4.20 pp, ten times its spread, and reducing to a single iteration also degrades significantly, so both the cross-modal path and the iteration itself contribute where coupling pays. The retrained full model (63.11) agrees with Table 1 (63.33) within seed noise.

On Soft Gating Removing the soft gate costs only 0.54 pp on VQA v2 (within the paired spread, Table 2), so the gate is not what makes the model work. It is, however, where the model can fail. On VQA v2 the learned gate collapsed to approximately 10^{-7} on real inputs, which makes the update $F(\boldsymbol{x}, \boldsymbol{z}_x, \boldsymbol{z}_y) = \boldsymbol{z}_x$ and drives both cross-modal Jacobians in Equation 5 to zero. This is the failure mode Section 2.1 identifies, observed in training.

Is the mixer the point? We replaced the cross-modal function M in Equation 6 with a gMLP (Liu et al., 2021) block, within two percent in parameter count (34.14M against 33.47M), and retrained on SNLI-VE. The two are indistinguishable: 75.26% against 75.15%, a seed-matched difference of $+0.11$ pp against a two-sigma spread of 0.25 pp. We also asked whether mixing again after the iteration helps, by replacing the pooled-concatenation readout with a symmetric cross-attention merge, against a capacity-matched control with the same parameter budget and no cross-modal path (38.196M against 38.195M). The difference between them is $+0.01$ pp. Neither the choice of mixer nor additional mixing after the iteration accounts for the

Table 2: Component ablation on VQA v2

Configuration	Acc (%)	Δ
Full MEQ	63.11	–
w/o Soft Gating	62.57	−0.54
w/o Cross-Attention	**58.90**	−4.20
w/o Self-Attention	62.33	−0.78
Single Iteration	62.39	−0.71

gain. This seems to hint at the architectural contribution to the current level of performance, but more study is needed to either confirm or refute this claim. We leave it as future work to further verify this.

Is it the two states? Two parameter-matched baselines replace only the state layout: one joint state that still carries tokens, and the feature-sum design of Ni et al. (2023), which pools each modality to a vector. MEQ leads both on VQA v2, by 0.35 pp and 11.39 pp against paired 2σ of 0.17 and 1.34 pp. CMU-MOSEI isolates the state count, since its state is already one vector per modality: collapsing the three costs 0.84 pp against a paired 2σ of 0.22 pp, even there, where MEQ does not beat its fusion baselines. On Hateful Memes nothing separates (Table 11). See Appendix C.3 ('State Design' paragraph) for more discussions on this matter.

Do these benchmarks need both modalities? CMU-MOSEI is the one benchmark where MEQ does not improve on its baselines, so we asked whether it rewards fusion at all. Retraining each model from scratch with one modality zeroed at its encoder output gives Table 3. On CMU-MOSEI a model that can see neither audio nor video is indistinguishable from the full model and from both baselines, and removing the cross-modal path from the update there costs 0.13 pp against a paired 2σ of 0.97 pp, where the same removal costs 4.20 pp on VQA v2. We therefore read the CMU-MOSEI result as a property of that benchmark under our feature pipeline rather than of the architecture. The same control separates cleanly on Hateful Memes and on SNLI-VE, so it is capable of detecting a modality that matters. SNLI-VE is worth a second look: its text-only model still reaches 70.42, far above the three-way chance rate, which is the annotation artifact SNLI is known for. On top of that artifact the image is worth 4.73 points to MEQ and only 1.19 to concatenation, so most of the 3.53 point gap between them in Table 1 is a difference in how much of the image each model uses.

Table 3: Single-modality controls. Each entry is a model retrained from scratch with one modality zeroed at the encoder output, means over three seeds. Metrics differ per benchmark, so only the verdicts are comparable across rows, not the sizes of the gaps. CMU-MOSEI has three modalities, so 'text only' there means audio and video both zeroed and an image-only arm does not apply.

Dataset	Metric	Both	Text only	Image only
SNLI-VE	Acc	75.15	70.42	33.85
Hateful Memes	AUROC	0.7080	0.6330	0.6526
CMU-MOSEI	Acc-7	54.70	53.97	–

4 Conclusion and Future Works

This work presents a new architecture based on mutual feedback to refine bimodal representations. Although we have mainly demonstrated our algorithm in the context of classification by fusion in the experiments, we would like to make it clear that the true contribution of this work goes beyond fusion. The real value of MEQ is in the iterative nature of information refinement while still offering performance better than, or on par with baselines. Under this viewpoint, we suggest several promising future directions:

- Multi-modal generalization: How to incorporate more than 2 modalities (Appendix B).
- Modular LLMs: Where multiple LLMs with different specializations iteratively refine each other's outputs.
- Agent agreement: Multiple agents converse to iteratively resolve discrepancy in observation.
- Iteration benefits: When and what problems benefit from MEQ?

In all of these cases, the key point that needs to be investigated is the theoretical foundation of iterative refinement under the given constraints.

Furthermore, the fact that MEQ is essentially a coupled dynamical system calls for closer investigation into the theoretical connections to traditional dynamical systems (e.g., Zames (1966)). A practical application in the same direction can also be found in Physics-Informed Neural Networks (Raissi et al., 2019), where PDEs that describe such physical problems are in fact dynamical systems.

AI Disclosure

In this work, we used generative AI tools for editing software code used in the experiments and relevant literature search. We have not used generative AI tools for any other tasks. All authors have reviewed all AI-assisted work. Specifically, the AI-generated code was reviewed and approved by all authors in-person, and the literature survey results were cross-checked as well. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

Every number in this paper regenerates from released run records: an anonymized bundle containing per-run configurations and results, model and training source, and scripts that rebuild each table byte-identically, with a verifier that checks file hashes and regenerates all tables from scratch. Checkpoints are indexed by SHA-256. The bundle is available for review at `https://anonymous.4open.science/r/mutual-equilibrium-5DCB`.

Ethics Statement

This work uses only datasets that are publicly available, and therefore does not require an IRB approval. No part of this work uses materials derived from human or animal subjects, either. Please see the 'AI Disclosure' statement above for AI tool uses in this work.

References

Donald G. M. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4), 1965.

Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. In *Proceedings of ICML*, 2019.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances of NeurIPS*, 2019.

Shaojie Bai, Vladlen Koltun, and J. Zico Kolter. Multiscale deep equilibrium models. In *Advances of NeurIPS*, 2020.

Shaojie Bai, Vladlen Koltun, and Zico Kolter. Stabilizing equilibrium models by jacobian regularization. In *Proceedings of ICML*, 2021.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 2019.

Charles G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92), 1965.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of CVPR*, 2021.

Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. COVAREP – A collaborative voice analysis repository for speech technologies. In *Proceedings of ICASSP*, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of ICML*, 2024.

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of CVPR*, 2020.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of EMNLP*, 2016.

Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. In *Advances of NeurIPS*, 2021.

Dimitris Gkoumas, Qiuchi Li, Christina Lioma, Yijun Yu, and Dawei Song. What makes the difference? An empirical comparison of fusion strategies for multimodal language analysis. *Information Fusion*, 66:184–197, 2021.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of CVPR*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances of NeurIPS*, 2020.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances of NeurIPS*, 2020.

Jiafan Li, Jiaqi Zhu, Liang Chang, Yuanzhe Li, Miaomiao Li, Yang Wang, and Hongfei Wang. Representation learning with mutual influence of modalities for node classification in multi-modal heterogeneous networks. In *Proceedings of IJCAI*, 2025.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to MLPs. In *Advances of NeurIPS*, 2021.

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient Low-rank Multimodal Fusion with modality-specific factors. In *Proceedings of ACL*, 2018.

Muhammad Arslan Manzoor, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang. Multimodality representation learning: A survey on evolution, pretraining and its applications. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 20(5), 2024.

Jinhong Ni, Yalong Bai, Wei Zhang, Ting Yao, and Tao Mei. Deep equilibrium multimodal fusion. *arXiv preprint arXiv:2306.16645*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of ICML*, 2021.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of ICCV*, 2017.

Mathieu Serrurier, Franck Mamalet, Thomas Fel, Louis Béthune, and Thibaut Boissin. On the explainable properties of 1-Lipschitz neural networks: An optimal transport perspective. In *Advances of NeurIPS*, 2023.

Sabrina Stöckli, Michael Schulte-Mecklenbeck, Stefan Borer, and Thomas Berger. Facial expression analysis with iMotions FACET platform: A validation study. *Behavior Research Methods*, 50(4): 1446–1460, 2018.

Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from Transformers. In *Proceedings of EMNLP*, 2019.

Ugo Tanielian and Gérard Biau. Approximating Lipschitz continuous functions with GroupSort neural networks. In *Proceedings of AISTATS*, pp. 442–450, 2021.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal Transformer for unaligned multimodal language sequences. In *Proceedings of ACL*, 2019.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-Agents enhances large language model capabilities. In *Proceedings of ICLR*, 2025.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

Jiayi Xin, Sukwon Yun, Jie Peng, Inyoung Choi, Jenna L. Ballard, Tianlong Chen, and Qi Long. $\texttt{I}^2\texttt{MoE}$: Interpretable multimodal interaction-aware Mixture-of-Experts. In *Proceedings of ICML*, 2025.

Peng Xu, Xiatian Zhu, and David A. Clifton. Multimodal learning with Transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12113–12132, 2023.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of CVPR*, 2019.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor Fusion Network for Multimodal Sentiment Analysis. In *Proceedings of EMNLP*, 2017.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proceedings ACL*, 2018.

George Zames. On the input-output stability of time-varying nonlinear feedback systems—part I: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11(2):228–238, 1966.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of CVPR*, 2019.

APPENDIX

A PROOFS AND DERIVATIONS

A.1 PROOF OF LEMMA 1

Lemma 1. *Given Equation 1, the Jacobians of $\boldsymbol{z}_x^*$ and $\boldsymbol{z}_y^*$ are given as*

$$\frac{d\boldsymbol{z}_x^*}{d(\cdot)} = \left(I - \frac{\partial F}{\partial \boldsymbol{z}_x^*} - \frac{\partial F}{\partial \boldsymbol{z}_y^*}\left(I - \frac{\partial G}{\partial \boldsymbol{z}_y^*}\right)^{-1}\frac{\partial G}{\partial \boldsymbol{z}_x^*}\right)^{-1}\left(\frac{\partial F}{\partial (\cdot)} + \frac{\partial F}{\partial \boldsymbol{z}_y^*}\left(I - \frac{\partial G}{\partial \boldsymbol{z}_y^*}\right)^{-1}\frac{\partial G}{\partial (\cdot)}\right)$$

$$\frac{d\boldsymbol{z}_y^*}{d(\cdot)} = \left(I - \frac{\partial G}{\partial \boldsymbol{z}_y^*} - \frac{\partial G}{\partial \boldsymbol{z}_x^*}\left(I - \frac{\partial F}{\partial \boldsymbol{z}_x^*}\right)^{-1}\frac{\partial F}{\partial \boldsymbol{z}_y^*}\right)^{-1}\left(\frac{\partial G}{\partial (\cdot)} + \frac{\partial G}{\partial \boldsymbol{z}_x^*}\left(I - \frac{\partial F}{\partial \boldsymbol{z}_x^*}\right)^{-1}\frac{\partial F}{\partial (\cdot)}\right),$$

where $(\cdot)$ is a placeholder indicating any independent variable (e.g., θ or $\boldsymbol{x}$).

Proof. For ease of presentation, we use the notation given in Equation 3. We start with the original fixed point equation given in Equation 1. Applying the implicit function theorem on both fixed points w.r.t. the placeholder $(\cdot)$ yields the following coupled equations (parameter subscripts omitted).

$$\frac{d\boldsymbol{z}_x^*}{d(\cdot)} = \frac{\partial F}{\partial (\cdot)} + F_{\boldsymbol{z}_x^*}\frac{d\boldsymbol{z}_x^*}{d(\cdot)} + F_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)} \tag{9}$$

$$\frac{d\boldsymbol{z}_y^*}{d(\cdot)} = \frac{\partial G}{\partial (\cdot)} + G_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)} + G_{\boldsymbol{z}_x^*}\frac{d\boldsymbol{z}_x^*}{d(\cdot)}. \tag{10}$$

We first tackle the $d\boldsymbol{z}_x^*/d(\cdot)$ by grouping like terms:

$$\frac{d\boldsymbol{z}_x^*}{d(\cdot)} = (I - F_{\boldsymbol{z}_x^*})^{-1}\left(\frac{\partial F}{\partial (\cdot)} + F_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)}\right) \tag{11}$$

Substitute this into Equation 10 to isolate the $d\boldsymbol{z}_y^*/d(\cdot)$ terms:

$$\begin{aligned}\frac{d\boldsymbol{z}_y^*}{d(\cdot)} &= \frac{\partial G}{\partial (\cdot)} + G_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)} + G_{\boldsymbol{z}_x^*}\left((I - F_{\boldsymbol{z}_x^*})^{-1}\left(\frac{\partial F}{\partial (\cdot)} + F_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)}\right)\right) \\ &= \frac{\partial G}{\partial (\cdot)} + G_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)} + G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}\frac{\partial F}{\partial (\cdot)} + G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}F_{\boldsymbol{z}_y^*}\frac{d\boldsymbol{z}_y^*}{d(\cdot)}\end{aligned}$$

Solving for $d\boldsymbol{z}_y^*/d(\cdot)$ yields the following.

$$\begin{aligned}\left(I - G_{\boldsymbol{z}_y^*} - G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}F_{\boldsymbol{z}_y^*}\right)\frac{d\boldsymbol{z}_y^*}{d(\cdot)} &= \frac{\partial G}{\partial (\cdot)} + G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}\frac{\partial F}{\partial (\cdot)} \\ \Rightarrow \frac{d\boldsymbol{z}_y^*}{d(\cdot)} &= \left(I - G_{\boldsymbol{z}_y^*} - G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}F_{\boldsymbol{z}_y^*}\right)^{-1}\left(\frac{\partial G}{\partial (\cdot)} + G_{\boldsymbol{z}_x^*}(I - F_{\boldsymbol{z}_x^*})^{-1}\frac{\partial F}{\partial (\cdot)}\right)\end{aligned}$$

Solving for $d\boldsymbol{z}_x^*/d(\cdot)$ is omitted due to symmetric arguments. □

Note that this derivation is assuming that $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent of $(\cdot)$. If that is not the case, simply change $\partial F/\partial(\cdot)$ and $\partial G/\partial(\cdot)$ to

$$\frac{\partial F}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial (\cdot)} \quad \text{and} \quad \frac{\partial G}{\partial \boldsymbol{y}}\frac{\partial \boldsymbol{y}}{\partial (\cdot)}, \text{ respectively.}$$

A.2 INFLUENCE DYNAMICS

Following the convention given in Equation 3, the Neumann series expansion of the influence of $d\boldsymbol{z}_x^*/d(\cdot)$ yields the following (The other influence term is omitted due to symmetry).

$$\left(I - F_{\boldsymbol{z}_x^*} - F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1}G_{\boldsymbol{z}_x^*}\right)^{-1} = \sum_{k=0}^{\infty}\left(\underbrace{F_{\boldsymbol{z}_x^*}}_{\text{Self-influence}} + \underbrace{F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1}G_{\boldsymbol{z}_x^*}}_{\text{Cross-modal influence}}\right)^k \tag{12}$$

Firstly, the self-influence term $F_{\boldsymbol{z}_x^*}$ indicates the direct influence of $\boldsymbol{z}_x^*$. Next, the latter cross-modality influence term indicates the influence interaction between F and G. To see why, notice the inner inverse term $(I - G_{\boldsymbol{z}_y^*})^{-1}$ is also an infinite sum $\sum_k (G_{\boldsymbol{z}_y^*})^k$, meaning the 'local' influence of $\boldsymbol{z}_y^*$ on G. It is being multiplied by $F_{\boldsymbol{z}_y^*}$ and $G_{\boldsymbol{z}_x^*}$ on both sides, which stand for single-step cross-modal influences. Put together, the term $F_{\boldsymbol{z}_x^*} + F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1} G_{\boldsymbol{z}_x^*}$ can be loosely interpreted as the amount of influence F processes on its own (term $F_{\boldsymbol{z}_x^*}$), *and* the amount of information F sends to G (term $G_{\boldsymbol{z}_x^*}$), that influence being combined and looped to refine $\boldsymbol{z}_y^*$ (term $(I - G_{\boldsymbol{z}_y^*})^{-1}$), and finally being sent back to F (term $F_{\boldsymbol{z}_y^*}$). Finally, this overall influence is being added over all possible powers (i.e., repetitions) of it. This means that the final inverse term is considering the information resulting from all possible influence paths.

A.3 Sensitivity

We wish to know how much the fixed point $\boldsymbol{z}^*$ gets perturbed if we perturb the input $\boldsymbol{u} = [\boldsymbol{x}; \boldsymbol{y}]$ by $\epsilon_{\boldsymbol{u}}$. To quantify this amount, we let $\boldsymbol{z}^*_{\boldsymbol{u}+\epsilon_{\boldsymbol{u}}} \triangleq \boldsymbol{z}^* + \epsilon_{\boldsymbol{z}}$ (i.e., how much the fixed point of $\boldsymbol{u} + \epsilon_{\boldsymbol{u}}$ deviates from the unperturbed fixed point $\boldsymbol{z}^*$.) By Taylor expansion, we have

$$
\begin{aligned}
T(\boldsymbol{z}^* + \epsilon_{\boldsymbol{z}}, \boldsymbol{u} + \epsilon_{\boldsymbol{u}}) &\approx T(\boldsymbol{z}^*, \boldsymbol{u}) + \frac{\partial T}{\partial \boldsymbol{z}^*}\epsilon_{\boldsymbol{z}} + \frac{\partial T}{\partial \boldsymbol{u}}\epsilon_{\boldsymbol{u}} \\
\Rightarrow \boldsymbol{z}^* + \epsilon_{\boldsymbol{z}} &\approx \boldsymbol{z}^* + \frac{\partial T}{\partial \boldsymbol{z}^*}\epsilon_{\boldsymbol{z}} + \frac{\partial T}{\partial \boldsymbol{u}}\epsilon_{\boldsymbol{u}} \\
\Rightarrow \|\epsilon_{\boldsymbol{z}}\| &\lesssim \left\| \left(I - \frac{\partial T}{\partial \boldsymbol{z}^*} \right)^{-1} \right\| \left\| \frac{\partial T}{\partial \boldsymbol{u}} \right\| \|\epsilon_{\boldsymbol{u}}\|.
\end{aligned}
$$

A.4 Sharper Condition for Invertibility

To ensure the invertibility of the total influence, we proposed to constrain the norm of the Jacobian J_T^*. That itself is sufficient, but we can derive a sharper condition at the expense of more computation.

Proposition 1. *If we have $\rho(G_{\boldsymbol{z}_y^*}) < 1$ and $\rho(F_{\boldsymbol{z}_x^*}) < 1$, the following is sufficient to guarantee the existence of the inverse term.*

$$
\frac{\|G_{\boldsymbol{z}_x^*}\| \|F_{\boldsymbol{z}_y^*}\|}{(1 - \|G_{\boldsymbol{z}_y^*}\|)(1 - \|F_{\boldsymbol{z}_x^*}\|)} < 1 \tag{13}
$$

Proof. Assume $\|G_{\boldsymbol{z}_y^*}\| < a \leq 1, \|F_{\boldsymbol{z}_x^*}\| < b \leq 1$ and factor the total influence term as follows:

$$
I - F_{\boldsymbol{z}_x^*} - F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1} G_{\boldsymbol{z}_x^*} = (I - F_{\boldsymbol{z}_x^*})(I - (I - F_{\boldsymbol{z}_x^*})^{-1} F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1} G_{\boldsymbol{z}_x^*}).
$$

Then the sufficient condition for the inverse of the LHS to exist is

$$
\rho((I - F_{\boldsymbol{z}_x^*})^{-1} F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1} G_{\boldsymbol{z}_x^*}) < 1. \tag{14}
$$

Instead of the spectral radius, which is hard to compute, let us focus on the norm since it forms an upper bound of ρ (the spectral radius is upper-bounded by the norm).
By the Neumann sum definition, we have

$$
\begin{aligned}
\left\|(I - F_{\boldsymbol{z}_x^*})^{-1}\right\| &\leq \sum_{k=0}^{\infty} \|F_{\boldsymbol{z}_x^*}\|^k < \frac{1}{1-b} \\
\left\|(I - G_{\boldsymbol{z}_y^*})^{-1}\right\| &\leq \sum_{k=0}^{\infty} \|G_{\boldsymbol{z}_y^*}\|^k < \frac{1}{1-a},
\end{aligned}
$$

due to the norm assumptions made above. Thus the bound of the norm of the LHS of Eqn. 14 is given as

$$
\left\|(I - F_{\boldsymbol{z}_x^*})^{-1} F_{\boldsymbol{z}_y^*}(I - G_{\boldsymbol{z}_y^*})^{-1} G_{\boldsymbol{z}_x^*}\right\| \leq \frac{\|G_{\boldsymbol{z}_x^*}\| \|F_{\boldsymbol{z}_y^*}\|}{(1-a)(1-b)} \tag{15}
$$

Inequality 15 ensures that the influence term is invertible. This implies that Eqn. 13 is sufficient condition for the invertibility of the influence term. □

Instead of bounding the norm of the entire J_T^*, this bound has the advantage that a fine-grained tuning is possible.

B More than Two Modalities

Although the main focus of this paper is to present bimodal feedback refinement due to theoretical tightness, we also mention a way to generalize this approach to more than two modalities. We propose two possibilities that will be left as future works.

Hierarchical composition Given modalities $\{\boldsymbol{x}_i\}_{i=1}^N$ as the input set, we form a merge tree $\{(((\boldsymbol{x}_i, \boldsymbol{x}_j), \boldsymbol{x}_k), \boldsymbol{x}_\ell), \cdots,)\}$ that dictates the order of combination. The ordering can be determined randomly, or preferably by domain knowledge. Then for each pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the MEQ generates the pair $(\boldsymbol{z}_i^*, \boldsymbol{z}_j^*)$. In order to apply it to the next $\boldsymbol{x}_k$, we need to combine the generated pair through another MLP $h : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$ to produce a temporary summary embedding $h(\boldsymbol{z}_i^*, \boldsymbol{z}_j^*)$. Then we take this summary and join with the next modality in line to produce the next embedding.
The shortcoming of this approach is obviously the order of mixing. Unless there is a justified way of ordering the modalities, random ordering will likely result in high-variance performance in the end. It also introduces an additional set of parameters for the combiner MLP.

Complete graph composition When all modalities need to interact with everyone else, hence forming a complete graph-like connection topology. In this case, the individual encoding function is defined as

$$\boldsymbol{z}_i^* = F_i(\boldsymbol{x}_i, \{\boldsymbol{z}_k\}_{k=1}^N).$$

This formulation is a natural extension of the idea given in this paper, and is intuitive to understand. Unfortunately, the standard Jacobian analysis we used in our work no longer yields simple solutions under this scheme. The Jacobian J_T^* now consists of N^2 block Jacobians, which means $(I - J_T^*)^{-1}$ no longer decomposes into nicely interpretable form given by Lemma 1. Hence, future works should further analyze the behavior of this larger Jacobian.

C Additional Experimental Results

C.1 Experimental Setups

Dataset details The following table shows the train-val-test split and class information in the datasets used. Following standard practice, we treat VQA v2 as classification over the 3,129 most frequent answers. Owing to compute constraints, we train and evaluate on the val2014 split, divided 9:1 into train and held-out portions with a fixed seed. The official train split is not used. The partition is over questions rather than images, and VQA v2 carries 5.29 questions per image on average, so the two halves share images: 99.7% of held-out questions use an image that also appears in training. The image encoder is frozen, so this exposure reaches the model only through the fusion block, but the VQA v2 numbers are best read as a held-out-question rather than a held-out-image estimate.

Table 4: Dataset statistics.

Dataset	Task	Train	Val	Test	Classes
Hateful Memes (Kiela et al., 2020)	Binary classification	8,500	500	1,000	2
VCR (Zellers et al., 2019)	4-way multiple choice	212K	26K	25K	4
VQA v2 (Goyal et al., 2017)	Open-ended QA	193K	21K	–	3,129
SNLI-VE (Xie et al., 2019)	Visual entailment	533K	9.6K	9.6K	3
CMU-MOSEI (Zadeh et al., 2018)	Sentiment (7-class)	16K	2K	5K	7

Implementation details We use a frozen CLIP (Radford et al., 2021) image encoder and a frozen BERT-base (Devlin et al., 2019) text encoder (per-benchmark variations in Table 6). The MEQ module has 768-dimensional hidden states with 8 attention heads. We iterate a damped fixed-point

update for $K = 10$ steps, $\boldsymbol{z}^{(k+1)} = \beta\, T(\boldsymbol{u}, \boldsymbol{z}^{(k)}) + (1 - \beta)\, \boldsymbol{z}^{(k)}$, and train with the Jacobian range penalty of Section 2.2 ($\rho_\ell = 0.7$, $\rho_h = 0.9$, Hutchinson estimate; β and probe counts in Table 5). The two mixing coefficients act at different levels. The mixing parameter α of the template in Equation 6 weighs the self-influence S against the cross-modal M within a block, while β damps the iteration of the joint operator, weighing the new iterate against the previous state.

Implicit differentiation would instead evaluate $(I - J_T^*)^{-1}$ at a root, whereas we read $\boldsymbol{z}^{(K)}$ at a nonzero residual (Table 9) and the undamped map is expansive on four of the five benchmarks (Table 12), so the Neumann series for that inverse does not converge there. All results are means over three seeds, reported on the validation (dev) split of each benchmark; for VCR we evaluate on the first 5,000 validation examples. CMU-MOSEI dataset is a tri-modal problem, for which we use a simple hierarchical approach: combine the image and audio modality first, then combine with text (Table 6).

Hyperparameters All runs use AdamW, a linear warmup of 5%, fixed-point tolerance 10^{-3}, and K=10 damped iterations.

Table 5: Training hyperparameters. $\sigma(d)$ denotes a learned scalar passed through a sigmoid, initialised at $d = 0.5$ so that β begins at 0.62; the other three benchmarks hold β fixed.

	SNLI-VE	Hateful	VCR	MOSEI	VQA v2
learning rate	1e-4	1e-4	1e-4	2e-5	1e-4
epochs	10	10	15	8	10
batch size	32	32	8	32	32
weight decay	0.01	0.01	0.01	1e-4	0.01
λ_j	0.5	0.5	0.5	0.5	0.5
λ_f	0.3	0.3	0.3	0.3	0
Hutchinson probes	5	1	1	5	1
damping β	$\sigma(d)$	0.5	0.5	$\sigma(d)$	0.5

Input encoders Table 6 is a summary of input encoders used in our experiments. All of our experiments use pretrained state-of-the-art encoders for both image and text modalities.

Table 6: Input encoders per benchmark (all frozen).

Dataset	Modality-1 encoder	Modality-2 encoder
SNLI-VE	CLIP image	BERT-base
Hateful Memes	CLIP image	BERT-base
VQA v2	CLIP image	BERT-base
VCR	CLIP image + RoI features	BERT-base
CMU-MOSEI	COVAREP (audio Degottex et al. (2014)) FACET (visual Stöckli et al. (2018))	BERT-base

C.2 Extra Task Results

Iterative refinement example We report an additional experiment on the SNLI-VE dataset to show that prediction accuracy increases over time. Figure 4 shows that accuracy rises steeply over the first three updates and is flat from k=5 onward, so the benefit comes from the early trajectory rather than from approaching a root.

Robustness Analysis We test robustness to input perturbation on the Hateful Memes dataset. The images were subjected to noise and blur perturbations while the texts were given typos. Table 7 shows the results over three seeds. The stochastic corruptions additionally average five independent draws, with all checkpoints seeing identical corrupted inputs within a draw.

Table 7: Robustness to input corruption on Hateful Memes (dev AUROC).

Perturbation	Concat.	MEQ
Image Perturbations		
Noise (σ=0.1)	0.6900	**0.7017**
Noise (σ=0.3)	0.6603	**0.6791**
Blur (r=2)	0.6850	**0.6923**
Blur (r=8)	**0.6582**	0.6564
Text Perturbations		
Typo (5%)	0.6621	**0.6791**
Typo (10%)	0.6508	**0.6623**

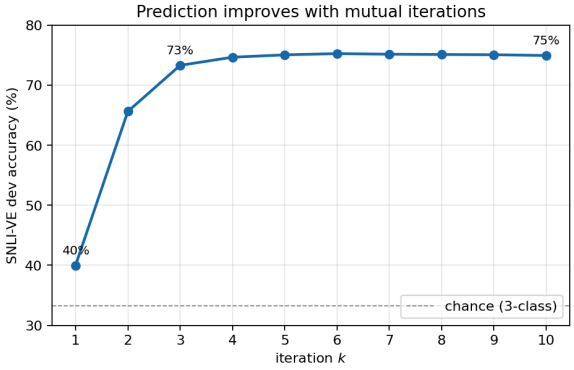


Figure 4: SNLI-VE dev accuracy read at each iteration k.

Under strong additive image noise MEQ leads by 1.88 pp (σ=0.3, paired 2σ of 1.51 pp over three seeds and five corruption draws).

COD10k results Direct training of MEQ and the baseline architectures on the camouflage benchmark of Figure 2. Results are averaged over three random seeds on a binary visual-entailment task constructed from COD10K images and automatically generated captions. Learning rates are chosen per architecture for training stability (3×10^{-5} for MEQ and concatenation, which otherwise collapses to a trivial optimum at 10^{-4}; 10^{-4} for the attention baselines and low-rank multimodal fusion, LMF (Liu et al., 2018)). All other hyperparameters follow Table 5.

Table 8: COD10k results.

	Concat	Cross-attn	Self-attn	LMF	MEQ
Accuracy	0.7495	0.7924	0.7849	0.7799	**0.7960**
Δ vs MEQ	−4.65pp	−0.35pp	−1.11pp	−1.61pp	–

Residual penalty effect Final dev residual and metric, means over three seeds. On VQA v2 the small residual under λ_f=0.3 is the identity collapse rather than convergence: seed 42 ends at 7.62×10^{-8} with the Jacobian norm at 1.000.

C.3 Analysis Experiments

When the residual penalty backfires On VQA v2 the residual term admits a degenerate solution. The identity map sets $\|r(\boldsymbol{z};\boldsymbol{u})\|$ to exactly zero, and the optimization found it: the learned gate closed to approximately 10^{-7} on real inputs, so no cross-modal information flowed. Setting $\lambda_f = 0$ on this

Table 9: Effect of the residual penalty.

	Residual		Metric	
Dataset	$\lambda_f = 0.3$	$\lambda_f = 0$	$\lambda_f = 0.3$	$\lambda_f = 0$
SNLI-VE	1.0e-2	1.0e-1	75.15	75.06
Hateful Memes	4.2e-2	7.4e-2	70.21	70.37
CMU-MOSEI	2.3e-2	4.4e-2	54.20	54.97
VCR	1.1e-2	6.7e-2	62.95	60.49
VQA v2	9.7e-4	9.4e-2	54.29	63.33

benchmark alone restores the coupling and improves accuracy by 9.04 pp (Table 9). The other four benchmarks keep $\lambda_f = 0.3$ (Table 5).

Can the collapse be prevented structurally? Disabling the penalty on one benchmark is a configuration choice rather than a fix, so we also ran the structural alternative: bounding that gate away from zero and leaving the penalty on. With a floor of 0.1 the constraint binds, since the gate settles at 0.1010 across seeds, and accuracy recovers 2.30 pp of the 9.04 pp gained by removing the penalty (Table 10). The coupling does not come back with it: the cross-modal to self-modal sensitivity ratio on the image path is 0.0027 under the floor against 0.0532 with the penalty off, and the penalty-free model settles at 0.2339, more than twice the floor it was never given. Bounding the gate closes one route to a zero residual while leaving another open, since the residual can also be driven down by shrinking the cross-modal function M itself. A safeguard acting on the gate alone is therefore not sufficient. Neither is a bound on the norm of the whole Jacobian: the collapsed run sits at $\|\hat{J}_T\| = 1.000$, comfortably above the lower end of the band of $\mathcal{L}_{jac}$ in Equation 8, while the off-diagonal blocks of Equation 3 are exactly what goes to zero. Penalising those blocks directly is what the analysis in Section 2.1 points to, so we ran that as well: a hinge holding each off-diagonal block above the value the penalty-free model reaches, with the residual penalty left on. It works as a mechanism. All three seeds finish with more cross-modal sensitivity than the penalty-free model has, measured on held-out batches with dropout disabled, at 1.9 to $15\times$ the floor. The accuracy does not follow: 53.35, which against the collapsed arm is -0.94pp on a paired 2σ of 1.52pp and is therefore not called. The two safeguards fail in opposite directions. Bounding the gate buys 2.30pp without restoring the coupling, and restoring the coupling buys nothing, so what the residual penalty costs this benchmark is not the cross-modal path itself.

Table 10: Structural safeguards against the identity collapse, on VQA v2. Means over three seeds. The last two columns are read at the epoch whose weights were kept, averaged over runs: the mean gate value on the image path, and that path's cross-modal sensitivity divided by its self-modal sensitivity.

Configuration	Acc (%)	Δ (pp)	called	gate	cross/self
$\lambda_f = 0.3$, no floor	54.29	–		–	–
$\lambda_f = 0.3$, gate floor 0.1	56.59	$+2.30$	yes	0.1010	0.0027
$\lambda_f = 0.3$, cross floor	53.35	-0.94	no	0.0894	0.0323
$\lambda_f = 0$	63.33	$+9.04$	yes	0.2339	0.0532

State design Each arm replaces only the state layout, with the iterated block widened until its parameter count matches. Matching is on the block: on VQA v2 the single-state arm ends up 18.9% larger in total, because widening the block also widens the projections feeding it and the head reading it, and it loses anyway. On Hateful Memes block-matching leaves that arm 12.7% larger, so a second arm matched on the whole model is reported; the two agree. On CMU-MOSEI the single-state arm is the block MEQ replaced, and MEQ's width was chosen by matching it, so all three blocks sit within 0.14% of each other by construction. On Hateful Memes the baseline is a MEQ retrained alongside these arms rather than the run behind Table 1. The feature-sum arms are that fusion design under our encoders, splits and budget, not a reproduction of Ni et al. (2023), and their numbers are not its reported scores.

On CMU-MOSEI the two manipulations separate: maintaining a state per modality is worth 0.84 pp, while letting those states read each other is not distinguishable from noise (0.13 pp against a paired 2σ of 0.97). Where the benchmark does reward fusion the cross-modal path is what matters instead, costing 4.20 pp on VQA v2 (Table 2). The CMU-MOSEI single-state arm also reaches a final residual smaller by a factor of 25 and still loses: converging better does not help here either, as in Figure 4.

Table 11: State-design comparison. Each arm replaces only the state layout, at matched block parameter count, under the recipe of Table 5. Means over three seeds; a difference is called when it exceeds twice the sample standard deviation of the paired per-seed difference. Metrics differ per benchmark, so only the verdicts are comparable across blocks, not the sizes of the gaps.

Dataset	Configuration	Score	Δ	called
VQA v2	MEQ (coupled states)	63.33	–	
	Single state, token level	62.98	-0.35	yes
	Feature-sum, one vector	51.94	-11.39	yes
Hateful Memes	MEQ (coupled states)	0.7094	–	
	Single state, block-matched	0.7257	$+1.63$	no
	Single state, model-matched	0.7262	$+1.68$	no
	Feature-sum, one vector	0.6905	-1.89	no
CMU-MOSEI	MEQ (coupled states)	54.70	–	
	Single state (replaced block)	53.86	-0.84	yes
	Feature-sum, one vector	53.99	-0.72	no

Table 12: Fixed-point behavior at the read depth $K{=}10$, measured on the full evaluation set of each benchmark for a single seed. On CMU-MOSEI n is $1{,}861$ of the $1{,}871$ validation clips: the text and the acoustic-visual features are distributed separately and ten clips fail to align.

				Residual		Metric		
Dataset	Seed	ρ at $\boldsymbol{z}^{(K)}$	Drift	at $\boldsymbol{z}^{(K)}$	at $\boldsymbol{z}^{(300)}$	at $\boldsymbol{z}^{(K)}$	at $\boldsymbol{z}^{(300)}$	n
CMU-MOSEI	42	0.965	0.38	3.2e-2	8.0e-6	53.9	51.8	1,861
VCR	42	1.011	0.60	1.6e-2	1.0e-3	62.8	55.5	5,000
Hateful Memes	43	1.083	1.05	3.3e-2	1.5e-3	70.7	56.6	500
VQA v2	42	1.340	1.42	7.4e-2	1.5e-3	62.8	27.4	21,029
SNLI-VE	43	1.689	4.51	1.1e-2	4.4e-3	75.0	53.2	9,602

Fixed-point behavior This experiment shows the per-dataset examinations of how the fixed-points behave (Table 12). ρ is the spectral radius of the undamped map and drift is $\|\boldsymbol{z}^{(300)} - \boldsymbol{z}^{(K)}\| / \|\boldsymbol{z}^{(K)}\|$. Only CMU-MOSEI is contractive at the read depth, and on every benchmark iterating far past K degrades the metric. This also empirically shows that it is beneficial to have a fixed point that can be reached in small number of iterations. The iteration does reduce the residual, however: it falls by a factor of three to eighty before the read depth is reached, from 0.81 to 1.1×10^{-2} on SNLI-VE and from 0.84 to 3.3×10^{-2} on Hateful Memes, and the residual columns show it continuing to fall past it. The decrease is monotone in k on four of the five; SNLI-VE is not monotone before $k{=}5$. The truncation therefore results in an approximate fixed point rather than an unrelated intermediate state.

What the iteration costs We measure the cost of the iteration at inference: forward pass only, batch 32, one RTX 3090, on the SNLI-VE architecture with random weights and random inputs (Table 13). At the read depth $K{=}10$ MEQ costs $2.13\times$ the fastest single-pass baseline in wall clock and $1.87\times$ in FLOPs. The marginal cost is flat: each iteration adds 1.687 GFLOPs per sample, constant to within 0.0003 across $K \in \{1, 2, 5, 10, 20\}$, on an encoder base that the sweep extrapolates to 19.204 GFLOPs against the 19.266 the concatenation baseline measures directly. Peak memory does not move with K, since the forward carries one state rather than a stack of K activations. Stopping early does not help here: left to run to a cap of 100 with the tolerance in charge, the iteration reaches the cap, which is the finding of Table 12 seen from the cost side.

Table 13: Inference cost at batch 32 on one RTX 3090, SNLI-VE architecture. Wall clock is the median of 30 timed passes after five warmups; FLOPs are counted for the whole forward including the frozen encoders; peak memory is measured per arm over an idle baseline. Random weights and random inputs, so this is the cost of a forward pass and not a statement about accuracy.

Model	ms / sample	GFLOPs / sample	peak MB
Concat MLP	1.465	19.27	1015
LMF	1.483	19.27	1003
Cross-attention	1.621	21.20	1004
Self-attention	1.745	21.96	1006
MEQ, $K{=}1$	1.627	20.89	1003
MEQ, $K{=}5$	2.304	27.64	1003
MEQ, $K{=}10$ (read depth)	3.127	36.08	1003
MEQ, $K{=}20$	4.778	52.95	1003