SynIB: Informational Bottleneck for Maximizing Synergy in Multimodal Learning

Konstantinos Kontras[1,3*] **Teodora Gagaleska**[1] **Thomas Strypsteen**[1]

Christos Chatzichristos[1] **Matthew Blaschko**[1] **Paul Pu Liang**[3] **Maarten De Vos**[1,2]

[1]Department of Electrical Engineering, KU Leuven, Leuven, Belgium
[2]Department of Development and Regeneration, KU Leuven, Leuven, Belgium
[3]MIT Media Lab and EECS, Cambridge, MA, USA

Abstract

A central objective in multimodal learning is to capture synergy: task-relevant information that arises only from the joint use of multiple modalities, and is not available from any single modality alone. While most approaches operate at the architectural level through larger or more complex fusion models, we propose a complementary axis: shaping the training objective itself. Standard training often emphasizes unimodal or redundant information, falling short on examples that require cross-modal reasoning. We formalize multimodal synergy through partial information decomposition (PID) and introduce the Synergistic Information Bottleneck (SynIB), a scalable, PID-inspired training objective oriented toward synergy. SynIB motivates the model to predict accurately from all modalities while penalizing confidence when information from any modality is withheld. Alongside the standard task loss, the model runs forward passes with one modality masked at a time and is penalized for remaining confident, which would indicate reliance on unimodal cues rather than cross-modal interactions. We validate SynIB in two regimes. On synthetic XOR tasks where the ground-truth synergy is known by construction, standard training fails to recover it while SynIB does. On six real-world benchmarks, including three MultiBench affective tasks, Hateful Memes and MM-IMDb with CLIP-ViT and DeBERTa backbones, and a controllable irony extension of CREMA-D that we introduce, SynIB improves accuracy on synergy-dependent examples by up to 6.4% and overall accuracy by up to 2.2%. Code: `github.com/kkontras/SynIB`.

1 Introduction

Multimodal learning aims to combine information from multiple sources to improve prediction. A central challenge is capturing synergistic information: predictive signal that arises from interactions between modalities rather than from any modality alone [Liang et al., 2024b]. In unimodal learning, nonlinear activations enable feature interactions and techniques such as weight decay [Krogh and Hertz, 1991] and dropout [Srivastava et al., 2014] reduce reliance on individual features [Goodfellow et al., 2016]; analogous mechanisms for cross-modal interactions remain underexplored.

Multimodal models sometimes over-rely on a single modality during training, favoring signals that yield faster optimization progress, a phenomenon known as multimodal competition [Huang

*Correspondence to: `kkontras@mit.edu`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

et al., 2022]. Numerous methods address this by estimating modality contributions and rebalancing learning through gradient modulation, auxiliary losses, or related interventions, but yield limited gains on synergistic cases [Kontras et al., 2025], indicating that correcting modality imbalance alone is insufficient to induce cross-modal synergy. A complementary line captures cross-modal interactions through architectural design, from fusion mechanisms [Zadeh et al., 2017, Tsai et al., 2019] to masked multimodal pretraining [Singh et al., 2022], expanding capacity for interaction without directly targeting synergistic prediction. Representation-level objectives such as contrastive alignment [Radford et al., 2021] do shape training, but pull modalities toward agreement, a property closer to redundancy than to synergy; masked contrastive pretraining [Wen et al., 2025] seeks synergy, but without the task label. What is missing across these lines is a supervised training signal that directly prioritizes synergistic prediction on the downstream task. Our approach fills this gap: it targets synergy at the loss level, is orthogonal to architectural and pretraining choices, and is complementary to rebalancing methods, which address a related but distinct failure mode.

To this end, we formalize multimodal synergy through partial information decomposition (PID) and introduce the Synergistic Information Bottleneck (SynIB), a PID-inspired, synergy-oriented training objective. The idea follows from the definition: synergistic information is the information that vanishes when any single modality is removed. A model that relies on synergy must therefore become uncertain when one modality is masked; a model that remains confident is relying on unimodal or redundant cues. SynIB turns this observation into a training signal by penalizing confident predictions under modality-wise masking, pressuring the model toward cross-modal dependencies on the examples where they matter most. Our main contributions are:

1. SynIB, to our knowledge, the first supervised, loss-level objective that targets synergy through counterfactual confidence penalties under modality-wise masking, requiring no prior knowledge of whether synergy exists in the data.
2. A motivating analysis showing that the failure to learn synergy is not caused by interference from unimodal signals; rather, synergistic cues are learned but quickly overfit, limiting generalization.
3. Extensive evaluation across real-world benchmarks from MultiBench, Hateful Memes, and MM-IMDb, including pretrained CLIP-ViT and DeBERTa backbones, a new irony extension of CREMA-D, and controlled Gaussian XOR tasks. On these SynIB improves synergy-dependent test set accuracy by up to 6.4% and overall accuracy by up to 2.2%.

2 Related Work

Partial Information Decomposition (PID). PID formalizes synergy, redundancy, and unique information across multiple sources with respect to a target [Williams and Beer, 2010, Griffith and Koch, 2014, Bertschinger et al., 2014]. In multimodal learning, it has primarily served as a post-hoc analytical tool to quantify information distribution in trained models [Liang et al., 2023], where synergy is operationalized as the predictive gap between a fusion model and an ensemble of unimodal models. Recent work incorporates PID-inspired quantities into training objectives via contrastive factorization [Liang et al., 2024a] and interaction-aware architectures [Xin et al., 2025]. Closest to ours, InfMasking [Wen et al., 2025] rewards agreement between masked and unmasked fused representations during self-supervised pretraining. SynIB is a complementary supervised objective that does the opposite: since synergy is the information lost when a modality is removed, it penalizes task predictions that survive masking (App. J).

Multimodal representation learning. A broad field of works captures cross-modal interactions through architectural and representational design: fusion mechanisms from tensor fusion [Zadeh et al., 2017] to cross-attention transformers [Tsai et al., 2019, Lu et al., 2019], alignment objectives such as multimodal contrastive learning [Radford et al., 2021], masked multimodal pretraining [Singh et al., 2022, Baevski et al., 2022], and factorized representations into modality-invariant and modality-specific subspaces [Liang et al., 2024a]. These approaches expand the capacity for cross-modal interaction, but none directly shape the training signal on the downstream task to penalize models that ignore synergistic cues, leaving even expressive architectures free to settle into unimodal or redundant solutions when those minimize loss faster [Huang et al., 2022]. Our objective is complementary: it operates at the loss level and can be combined with any of these architectural choices.

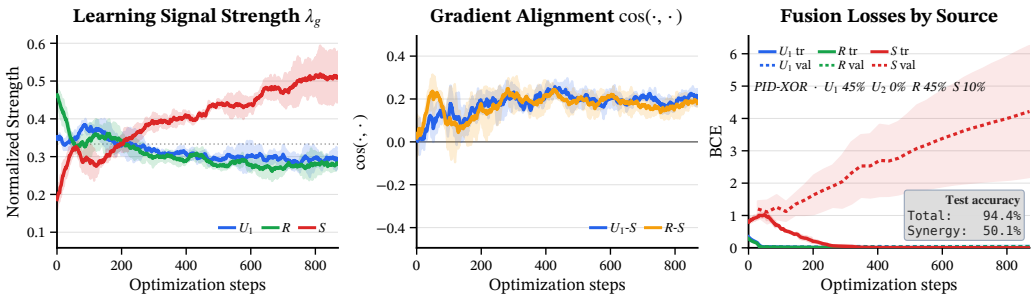


Figure 1: **Gradient geometry across PID sources under vanilla fusion.** A model is trained on examples drawn from the three PID sources, U_1, R, and S, with $U_2 = 0$ by construction (details in Sec. 4.2). **Left:** Per-group learning signal strength is substantial for all sources, meaning that the examples of that source create gradient capable of changing the parameters, with synergistic examples producing the largest λ_g. **Center:** Gradient alignment between source pairs stays near zero or slightly positive throughout training, indicating no destructive competition. **Right:** Fusion BCE by source on intact inputs (solid train, dashed validation). Synergistic training loss drops to zero while validation rises, indicating overfitting on the scarce synergistic examples and leaving test synergy accuracy at chance (50.1%; total 94.4%).

Multimodal competition. A complementary line of work addresses cross-modal learning failures as multimodal competition, estimating modality contributions and rebalancing learning accordingly. Strategies include measuring contribution via unimodal performance [Yao and Mihalcea, 2022, Peng et al., 2022, Vielzeuf et al., 2018, Kontras et al., 2024, Fan et al., 2023] and enforcing balance through gradient modulation or decomposition [Wei and Hu, 2024], reinitializing modality-specific components to escape suboptimal trajectories [Wei et al., 2024], perturbation-based attribution such as Shapley approximations [Li et al., 2023], permutation importance [Kontras et al., 2025], counterfactual modality removal [Ji et al., 2022], and ensemble-based strategies that sidestep fusion [Hua et al., 2024]. While effective at preventing modality collapse, these methods assume balanced contributions suffice for multimodal learning. As MCR [Kontras et al., 2025] shows, however, correcting modality imbalance does not by itself induce synergy.

3 Method

Problem setup. We consider a supervised multimodal prediction task with two input modalities X_1 and X_2 and a target Y: $(X_1 \in \mathbb{R}^{d_1},\ X_2 \in \mathbb{R}^{d_2},\ Y \in \mathcal{Y}) \sim p(x_1, x_2, y)$. Modality-specific encoders $f_{\theta_i} : \mathbb{R}^{d_i} \to \mathbb{R}^{m_i}$ produce $Z_i = f_{\theta_i}(X_i)$ for $i \in \{1, 2\}$, a fusion network $f_{\theta_{12}} : \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \Delta(\mathcal{Y})$ it's multimodal outputs $\hat{Y}_{12} = f_{\theta_{c_{12}}}(Z_1, Z_2)$, and the unimodal encoders $f_{\theta_{c_i}} : \mathbb{R}^{m_i} \times \mathbb{R}$ the unimodal outputs $\hat{Y}_i = f_{\theta_{c_i}}(Z_i)$. We collect all trainable parameters as $\theta = \{\theta_1, \theta_2\, \theta_{c_1}, \theta_{c_2}, \theta_{c_{12}}\}$.

Synergy as a learning goal. Partial Information Decomposition [Williams and Beer, 2010, Bertschinger et al., 2014] decomposes the predictive mutual information into four atoms:

$$I(Y; X_1, X_2) \;=\; R(Y; X_1, X_2) \;+\; U_1(Y; X_1 \mid X_2) \;+\; U_2(Y; X_2 \mid X_1) \;+\; S(Y; X_1, X_2), \quad (1)$$

where R is redundant across modalities, U_i is unique to X_i, and S is accessible only from the joint (X_1, X_2). Following Bertschinger et al. [2014], synergy is the gap between $I_p(Y; X_1, X_2)$ and its minimum over distributions preserving the bivariate marginals (Y, X_1) and (Y, X_2):

$$S(Y; X_1, X_2) = \underbrace{I_p(Y; X_1, X_2)}_{\substack{\text{predictive information} \\ \text{in the joint distribution}}} - \underbrace{\min_{q \in \Delta_p} \int q(y, x_1, x_2) \log \frac{q(y \mid x_1, x_2)}{q(y)}\, dy\, dx_1\, dx_2}_{\text{maximum predictive information from the unimodal marginals}}, \quad (2)$$

with $\Delta_p = \{\, q : q(y, x_i) = p(y, x_i),\ i \in \{1, 2\}\,\}$. The first term is the predictive information available under the joint distribution; the second upper-bounds what any predictor whose dependence on (X_1, X_2) factors through the bivariate marginals. Standard cross-entropy training maximizes $I(Y; X_1, X_2)$ as a single scalar but offers no mechanism to allocate gradient signal to S rather than R, U_1, or U_2 [Liang et al., 2023], motivating an objective that prioritizes S.

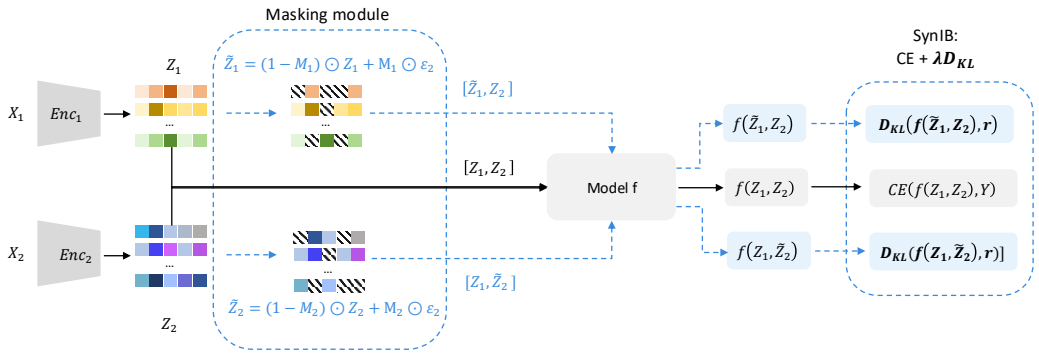


Figure 2: **SynIB overview.** Standard multimodal fusion (black) trains a model to predict Y from (Z_1, Z_2), leaving optimization free to settle on unimodal or redundant cues. SynIB (blue) adds counterfactual passes in which one modality is replaced with a feature-masked version $\tilde{Z}_i$ that removes its task-relevant content, and penalizes the model when its predictions remain confident under this corruption. Confidence under masking signals reliance on a single modality; the KL penalty against a reference distribution r pushes the model toward predictions that depend on the joint (Z_1, Z_2), so the cross-entropy objective prioritizes learning from cross-modal interactions.

3.1 A Motivating Analysis: What Makes Synergy Hard to Learn?

A common hypothesis attributes the difficulty of learning synergy to gradient interference between modalities or PID sources [Wei and Hu, 2024]. We probe this on a bimodal XOR with single-source examples (App. F.3), attributing gradient signal to specific components.

Observations. We diagnose vanilla training using two per-source quantities derived from the empirical Neural Tangent Kernel (NTK) [Jacot et al., 2018, Chizat et al., 2019], which characterizes training dynamics through inner products of per-example gradients: the *learning signal strength* λ_g, measuring how strongly examples from PID source g drive parameter updates, and the *gradient alignment* $\cos(g, h)$, measuring interference between sources (derivations in App. K). Figure 1 shows synergistic examples receiving the largest λ_g (left) and pairwise alignments staying near zero (center): synergy is neither starved of gradient signal nor in destructive competition with other sources. Yet its training loss drops while its validation loss rises (right). A large λ_g speeds up fitting but does not ensure generalization: the model *fits* the synergistic training examples without *learning* the cross-modal rule that generates them, memorizing this scarce subset instead of extracting an interaction that neither modality reveals alone.

Synergy fails at generalization. The bottleneck is not gradient signal; capacity helps (Table 1a) but does not decide which solution training selects for the synergistic examples. SynIB is designed to steer this selection toward the cross-modal interaction; App. A.1 reports the same diagnostics under SynIB.

3.2 The SynIB Training Objective

Section 3.1 shows that vanilla training does reduce the training loss on synergistic examples, but through a shortcut: it memorizes them from features available within a single modality instead of learning the cross-modal interaction that would generalize. SynIB addresses this by penalizing the model when it remains confident under a modality corruption $\tilde{X}_1$ (with latent $\tilde{Z}_1 = f_{\theta_1}(\tilde{X}_1)$): confidence under the counterfactual $(\tilde{X}_1, X_2)$ indicates the model reached its answer without cross-modal interaction, pushing it toward joint use of both modalities. The objective combines cross-entropy on intact inputs with a confidence penalty on counterfactuals.

The remainder of this section develops each component. Sec. 3.3 grounds the confidence penalty in information-theory; Sec. 3.4 derives a tractable variational surrogate; Sec. 3.5 specifies the corruption operator $\tilde{X}_1$; and Sec. 3.6 assembles the full objective.

3.3 Quantifying Complementarity from Counterfactual Predictions

The confidence penalty under corruption measures how much the model's predictions change when information from one modality is removed. This quantity has a principled interpretation, derived from the information X_1 contributes to predicting Y beyond what is available from X_2 alone.

The information X_1 carries about Y given X_2 is captured by the conditional mutual information $I(X_1; Y \mid X_2) = H(Y \mid X_2) - H(Y \mid X_1, X_2)$. Computing this quantity requires marginalizing X_1 out of $H(Y \mid X_2)$, which is intractable in high dimensions. We replace this marginalization with a single targeted perturbation $\tilde{X}_1$ designed to remove X_1's task-relevant content, and measure the resulting loss in conditional predictability, the *Mutual Information Perturbation Difference*:

$$\mathrm{MIPD}_1 = H(Y \mid \tilde{X}_1, X_2) - H(Y \mid X_1, X_2) = I(X_1; Y \mid X_2) - I(\tilde{X}_1; Y \mid X_2) \leq I(X_1; Y \mid X_2). \tag{3}$$

Maximizing MIPD_1 maximizes a lower bound on the contribution of X_1 beyond X_2; derivation and tightness conditions are in App. B. This transforms the question of estimating CMI into the design problem of constructing $\tilde{X}_1$, addressed in Sec. 3.5.

3.4 Variational Approximation for Tractable Cross-Modal Optimization

Eq. (3) is not directly optimizable: the conditional entropies require access to the true predictive distributions $p(y \mid x_1, x_2)$ and $p(y \mid \tilde{x}_1, x_2)$. We obtain a tractable surrogate by approximating each entropy through variational quantities the model can compute.

For $H(Y \mid X_1, X_2)$, a variational decoder $q_\theta(y \mid x_1, x_2)$ and the non-negativity of KL [Barber and Agakov, 2004] give the standard upper bound $H(Y \mid X_1, X_2) \leq \mathbb{E}_{p(x_1, x_2, y)}[-\log q_\theta(y \mid x_1, x_2)]$, recovering the cross-entropy loss. For $H(Y \mid \tilde{X}_1, X_2)$, the reference-based entropy identity [Tishby et al., 2000, Alemi et al., 2016] gives

$$H(Y \mid \tilde{X}_1, X_2) = C_r - \mathbb{E}_{p(\tilde{x}_1, x_2)} D_{\mathrm{KL}}(p(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot)), \tag{4}$$

where $C_r = \mathbb{E}_{p(y)}[-\log r(y)]$ is constant in θ. Substituting q_θ for p in Eq. (4) defines the variational surrogate $\widehat{H}_\theta(Y \mid \tilde{X}_1, X_2)$, which recovers the true entropy when $q_\theta = p$. Combining $\widehat{H}_\theta$ with the cross-entropy upper bound on $H(Y \mid X_1, X_2)$ gives a tractable surrogate for MIPD_1, which coincides with the true value when the variational family contains p. The two terms together form the SynIB training objective, assembled in Sec. 3.6. Variational families, closed-form KL divergences, and properties of the surrogate are in Appendices C and B.

3.5 Counterfactual Corruption via Feature Masking

The MIPD lower bound from Sec. 3.3 tightens with smaller $I(\tilde{X}_1; Y \mid X_2)$, giving a concrete design criterion: choose $\tilde{X}_1$ to minimize residual task-relevant information while perturbing as little of X_1 as possible. We implement the corruption as feature-wise masking through a binary mask $M_1 \in \{0, 1\}^{d_1}$ and a gate $\tilde{z}_1 = (1 - M_1) \odot z_1 + M_1 \odot \epsilon$, where ϵ is noise independent of Y and coordinates with $M_{1,j} = 1$ are corrupted. Feature-level control gives more precise residual minimization than global perturbations, and the operator can be applied equivalently to x_i or the latent z_i.

The choice of M_1 determines what SynIB ends up penalizing. The ideal choice would corrupt cross-modal cues while leaving unimodal ones intact, so that any remaining confidence under $(\tilde{X}_1, X_2)$ comes from unimodal reliance and gets penalized. Since unimodal cues are not labeled, we identify them adversarially: a **learned mask** $M_{\psi_1} \in \{0, 1\}^{d_1}$ is trained to flag the minimum set of coordinates whose corruption collapses unimodal prediction,

$$\mathcal{L}_{\text{mask}}(\psi_1) = -\mathrm{CE}\big(f_{\theta_{c_1}}((1 - M_{\psi_1}) \odot x_1 + M_{\psi_1} \odot \epsilon),\, y\big) + \lambda_M \|M_{\psi_1}\|_1, \tag{5}$$

and the SynIB corruption mask is its inverse, $M_1 = \mathbf{1} - M_{\psi_1}$. This isolates the features the unimodal predictor relies on: SynIB preserves them and corrupts the rest, leaving the model with only unimodal cues at the counterfactual pass. Eq. (5) is equivalent to finding the sparsest perturbation that maximally degrades the unimodal log-likelihood (App. E). It assumes only that the heads rely on a sparse set of coordinates, not that information types occupy separate ones (App. F.3.3).

Alongside learned masking, we examine a simpler **random masking** variant, $M_1 \sim \mathrm{Bernoulli}(\pi)$, which corrupts coordinates independently without consulting any predictor. This variant has no

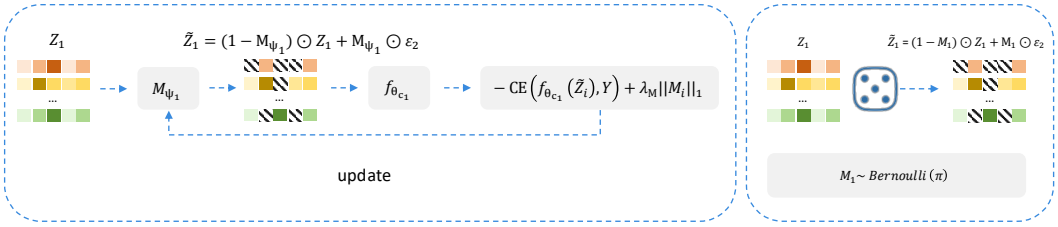


Figure 3: Two strategies for constructing the counterfactual mask M. **Left:** Learned masking trains M_{ψ_1} adversarially against a unimodal predictor with a sparsity penalty, isolating the features needed for unimodal prediction; SynIB inverts this mask so the corruption removes everything else, leaving only unimodal cues intact. **Right:** Random masking samples each coordinate independently as $M_i \sim \text{Bernoulli}(\pi)$, corrupting features without assumptions about their content.

learned parameters and provides a strong task-agnostic baseline when learned masking is impractical, and it isolates the benefits of precise mask construction. Figure 3 illustrates the two constructions.

3.6 The SynIB Objective

Combining the surrogate (Sec. 3.4) with the masking operator (Sec. 3.5) yields the SynIB objective:

$$\mathcal{L} = \underbrace{\mathbb{E}_{p(x_1,x_2,y)}[-\log q_\theta(y \mid x_1, x_2)]}_{\text{accurate prediction with intact modalities}} + \lambda \underbrace{\mathbb{E}_{p(x_1,x_2)}\mathbb{E}_{p(\tilde{x}_1|x_1)} D_{\mathrm{KL}}(q_\theta(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot))}_{\text{uncertainty under modality corruption}} \tag{6}$$

where $q_\theta(y \mid x_1, x_2)$ is the model's predictive distribution, $r(y)$ a fixed reference distribution (uniform by default; App. C.3), and $\lambda \in \mathbb{R}^+$ the regularization strength. Figure 2 illustrates the resulting training procedure: at each step, the fusion model runs forward passes on the intact pair and on each counterfactual, with the cross-entropy term applied to the first and the KL penalty to the others.

More than two modalities. With n modalities, the corruption applies to a subset $A \subset \{1, \ldots, n\}$, and Eq. (3) becomes $\mathrm{MIPD}_A = I(X_A; Y \mid X_{-A}) - I(\tilde{X}_A; Y \mid X_{-A})$: the information about Y lost with X_A that the remaining modalities X_{-A} cannot supply. SynIB sums the KL term of Eq. (6) over a family $\mathcal{A}$ of masked subsets; with two modalities, $\mathcal{A} = \{\{1\}, \{2\}\}$. Masking single modalities can miss information that two modalities hold redundantly; masking pairs exposes it, and at $n = 3$ the two cover all six proper subsets.

4 Experiments

We evaluate SynIB on six real-world multimodal benchmarks (Sec. 4.1), comprising three MultiBench affective tasks, Hateful Memes and MM-IMDb with large pretrained backbones, and a controllable irony extension of CREMA-D we introduce, and on synthetic XOR tasks (Sec. 4.2). We ask whether SynIB improves performance on examples requiring cross-modal reasoning, and whether targeting synergy harms accuracy on the full test set where unimodal cues already suffice.

4.1 Real-World Multimodal Benchmarks

We evaluate on six real-world multimodal tasks. Three come from MultiBench [Liang et al., 2021]: UR-Funny [Hasan et al., 2019] (humor detection), MUStARD [Castro et al., 2019] (sarcasm detection), and CMU-MOSI [Zadeh et al., 2016] (sentiment analysis). The fourth is Hateful Memes [Kiela et al., 2020], where hatefulness frequently emerges only from the joint interpretation of image and text. The fifth is MM-IMDb [Arevalo et al., 2017], movie-genre classification from posters and plot summaries (five most frequent genres), outside affective computing; its two modalities are about equally strong (73.6 and 74.3 unimodal accuracy) and err largely independently. The sixth is CREMA-D [Cao et al., 2014], a six-class audio–visual emotion recognition benchmark whose largely congruent signals admit strong unimodal performance. Because CREMA-D does not stress cross-modal integration, we further construct CREMA-D-Irony: for a fraction α of examples we replace the audio with a donor clip carrying a contradicting emotion and relabel the sample as *ironic*. The rate α controls synergy density (App. H).

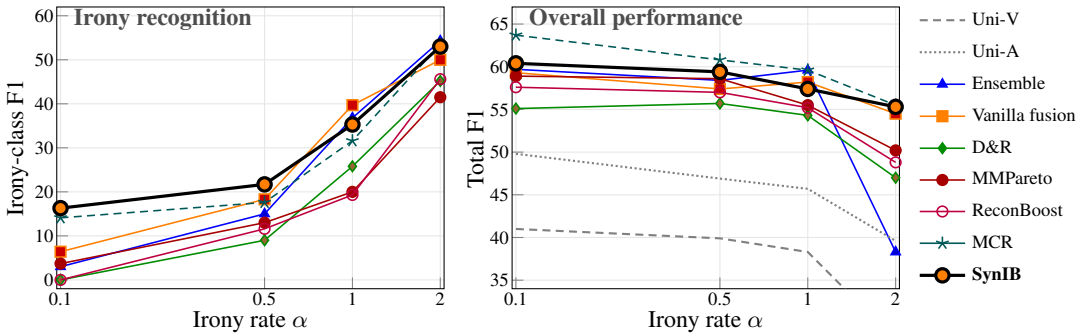


Figure 4: F1 scores on the CREMA-D irony recognition task under varying irony rates α; SynIB uses random masking ($\pi = 0.3$). **Left:** irony-class F1. **Right:** total F1. Standard fusion and prior balancing methods yield unstable gains when irony is rare, whereas SynIB gives the best irony F1 for $\alpha \leq 0.5$ and stays within 3.3 points of the best total F1.

Baselines. We compare SynIB against unimodal models, late ensembling, vanilla fusion, and four modality-balancing methods: (1) D&R [Wei et al., 2024], which periodically re-initializes encoders; (2) MMPareto [Wei and Hu, 2024], which balances unimodal and multimodal gradients via a Pareto rule; (3) ReconBoost [Hua et al., 2024], which alternates unimodal updates with KL reconciliation; and (4) MCR [Kontras et al., 2025], which adds a mutual-information regularizer to the fusion loss. We also include CE+InfMasking, which adds the self-supervised InfMasking loss [Wen et al., 2025] to cross-entropy with a tuned weight (MultiBench, Hateful Memes, and MM-IMDb; App. J). All methods share the same backbone and differ only in the training objective (full descriptions in App. J). For Hateful Memes and MM-IMDb we use CLIP ViT-B/16 [Radford et al., 2021] (86M) and DeBERTa-v3-base [He et al., 2021] (184M) encoders, feeding a fusion Transformer on Hateful Memes. SynIB M_{Random} uses $\pi = 0.7$ on UR-Funny, 0.1 on MUStARD, and 0.3 on MOSI, Hateful Memes, MM-IMDb, and CREMA-D-Irony (sensitivity in App. I.4).

Isolating synergy-dependent examples. Frequently multimodal datasets test examples are solvable from a single modality. To isolate cases requiring multimodal integration, we define a **synergy subset** of test samples misclassified by every unimodal model. The subset size indicates each task's reliance on cross-modal interaction: Hateful Memes (26.5%), MUStARD (12.7%), UR-Funny (11.6%), MM-IMDb (8.0%), and MOSI (2.5%). With only 17 examples, the MOSI synergy subset is too small for firm conclusions; we treat MOSI as a check that synergy-targeted training does not harm tasks where unimodal cues already suffice. CREMA-D-Irony complements this post-hoc view: its irony class is constructed to require cross-modal integration, with α controlling its density, and 90.7% of its test samples fall in our synergy subset. App. I.1 repeats the comparison under six alternative definitions, including a model-free one.

Results on CREMA-D-Irony. Figure 4 reports irony-class F1 (left) and total F1 (right) across α. SynIB outperforms MCR, the strongest balancing baseline, on irony F1 for $\alpha \leq 1$, most clearly at $\alpha = 0.5$ (21.7 vs. 17.6), and is within 0.5 points at $\alpha = 2.0$. Vanilla fusion and ensembling become competitive only when synergy is abundant ($\alpha \geq 1$). SynIB stays within 0.2–3.3 points of the best total F1 at every α (full numbers in App. H.7).

Results on MultiBench, Hateful Memes, and MM-IMDb. Figure 5 reports accuracy on the synergy subset and on the full test set (paired significance tests in App. I.6). On the synergy subset, SynIB achieves the largest gains across all five benchmarks, with learned masking improving over the strongest baseline by $+6.4$ on MUStARD, $+3.6$ on UR-Funny, $+1.1$ on Hateful Memes, and $+0.7$ on MM-IMDb; random masking is strongest on Hateful Memes ($+3.0$) but more variable elsewhere, suggesting that the targeted mask is what reliably drives improvements on synergy-required examples. Modality-balancing methods (D&R, MMPareto, ReconBoost, MCR) yield inconsistent gains on the synergy subset, consistent with our argument that rebalancing alone does not induce synergistic learning. CE+InfMasking is similarly inconsistent, falling below vanilla fusion on UR-Funny, MOSI, and MM-IMDb, as expected from an objective that does not isolate synergy (Sec. 2); both SynIB variants exceed it on all five synergy subsets. On the full test set, SynIB remains

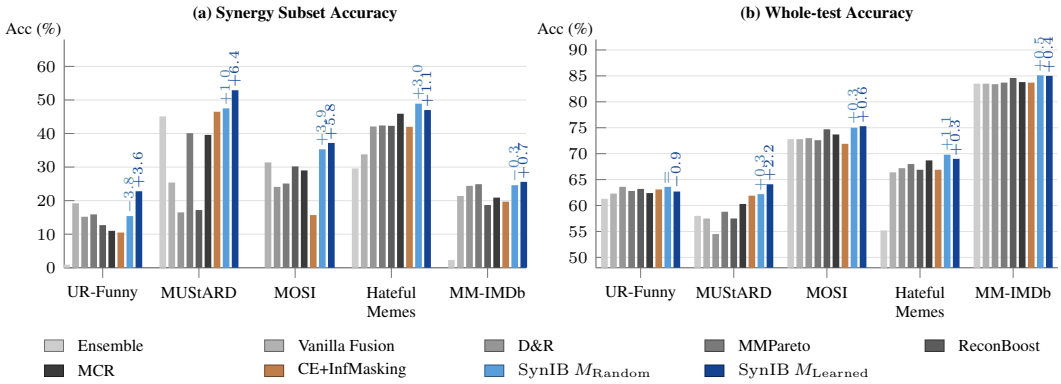


Figure 5: Comparison across real-world multimodal benchmarks. **(a)** Synergy-subset accuracy on UR-Funny, MUStARD, MOSI, Hateful Memes, and MM-IMDb (samples misclassified by every unimodal model). **(b)** Whole-test accuracy. Annotations above SynIB bars give the difference relative to the best baseline. SynIB consistently improves synergy-subset accuracy while remaining competitive on the full test set.

Table 1: Synergy-subset (Syn.) and whole-test (All) accuracy, mean ± sd over 3 runs; bold: best per column. **(a)** Hateful Memes, larger and pretrained backbones (base setting in Fig. 5). **(b)** UR-Funny, three modalities; SynIB masks single modalities, pairs, or both (Sec. 3.6); AGM [Li et al., 2023] is an extra baseline.

(a)

	Hateful Memes (image, text)					
	CLIP-L + DeBERTa-L 740M		CLIP-L + Qwen2.5 1.8B		FLAVA 241M	
	Syn.	All	Syn.	All	Syn.	All
Vanilla	53.4 ±3.5	72.1 ±1.5	53.1 ±4.6	72.2 ±1.9	32.5 ±16.3	61.8 ±4.9
D&R	58.0 ±1.6	73.4 ±1.0	53.4 ±3.2	70.6 ±0.6	35.8 ±2.8	62.7 ±1.1
MMPareto	55.6 ±2.8	72.0 ±0.2	54.5 ±6.8	72.2 ±1.5	40.0 ±2.1	**65.6** ±0.9
ReconBoost	51.9 ±11.4	71.6 ±2.6	53.3 ±1.1	71.1 ±0.6	36.6 ±5.7	62.6 ±3.2
MCR	55.3 ±4.5	72.3 ±1.1	**58.0** ±4.8	**73.9** ±0.8	31.5 ±2.6	62.4 ±1.4
SynIB M_{R}	**59.1** ±5.2	**74.0** ±0.6	56.5 ±3.3	72.7 ±1.8	**43.4** ±5.8	65.1 ±2.1
SynIB M_{L}	58.6 ±4.2	72.7 ±1.2	56.4 ±6.6	71.8 ±1.4	41.3 ±3.0	64.5 ±1.3

(b)

	UR-Funny	
	three modalities text, audio, video	
	Syn.	All
Vanilla	22.6 ±4.6	61.6 ±1.0
MCR	22.7 ±10.5	62.5 ±2.0
AGM	15.5 ±13.8	62.9 ±1.0
MMPareto	21.3 ±8.7	62.3 ±2.4
SynIB, singles	**25.3** ±4.5	63.2 ±1.1
SynIB, pairs	23.2 ±1.4	63.3 ±1.3
SynIB, both	24.4 ±2.7	**64.3** ±1.2

the top performer on Hateful Memes (+1.1 random, +0.3 learned), MM-IMDb (+0.5 random, +0.4 learned), and MUStARD (+2.2 learned), and stays within 1 point on UR-Funny (−0.9) and MOSI (+0.6), indicating no cost to overall accuracy.

Larger and pretrained backbones. On Hateful Memes, scaling the encoders to CLIP-L/14 with DeBERTa-v3-large (740M parameters) or Qwen2.5-1.5B (1.8B) raises vanilla fusion's synergy-subset accuracy from 33.8 to 53.4 and 53.1, so capacity helps. SynIB still improves on it at both scales (59.1 and 56.5) and ranks first at 740M, while MCR leads at 1.8B (58.0). The most direct test replaces our fusion Transformer with FLAVA, whose fusion encoder is itself pretrained for cross-modal interaction: there SynIB raises synergy-subset accuracy from 32.5 to 43.4, 3.4 points above the best baseline (Table 1a; setup in App. I.5).

Three modalities. On UR-Funny with all three modalities, every choice of masked subsets improves over vanilla fusion and the balancing baselines on both metrics (Table 1b); the three choices differ by less than the seed spread, and masking all subsets is the safer default, as it cannot miss redundantly held information.

Across six benchmarks spanning humor, sarcasm, sentiment, hate speech, movie genres, and emotion-irony, SynIB consistently improves performance on the examples that demonstrably require cross-

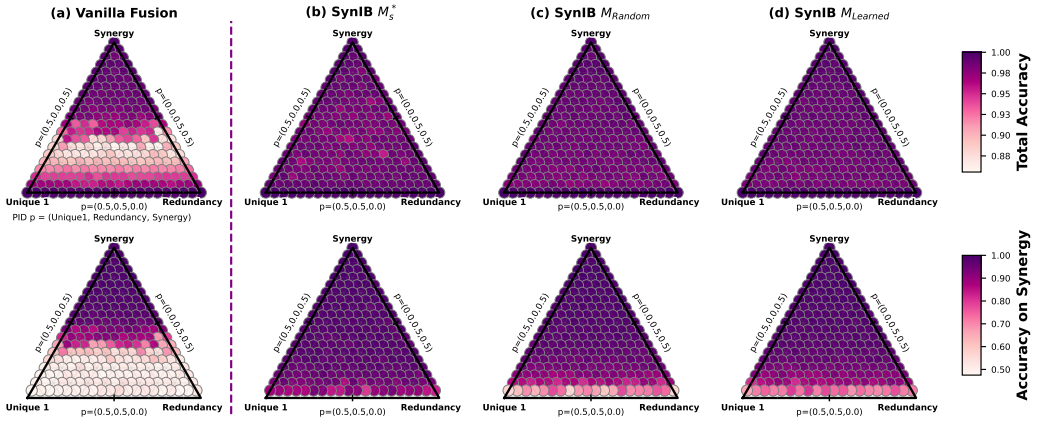


Figure 7: **Performance across PID-controlled data regimes on the synthetic XOR task.** Each triangle is a probability simplex over information types $(p_{\mathrm{U1}}, p_{\mathrm{Red}}, p_{\mathrm{Syn}})$; top row shows total accuracy, bottom row accuracy on synergistic examples only. Vanilla fusion (a) degrades sharply as synergy dominates. SynIB with oracle masking (b) resolves the task uniformly. Random masking (c) provides inconsistent gains, while learned masking (d) closely recovers oracle performance.

modal reasoning, while remaining competitive, and often best, on the full test distribution. The pattern is most pronounced on tasks with substantial synergy content (CREMA-D-Irony, MUStARD, Hateful Memes) and naturally smaller on tasks where unimodal shortcuts already solve most of the problem.

4.2 Synthetic Experiments

We design two bimodal XOR tasks. *Spurious XOR* probes resistance to unimodal shortcuts: the label is the XOR of two modality-specific bits (neither modality alone informative), but one modality additionally carries a label-correlated feature available only at training, letting the model fit training perfectly without the cross-modal signal. *PID-Controlled XOR* sweeps the density of each PID source (unique-to-X_1, redundant, synergistic). Full setup in App. F.

Robustness to spurious shortcuts. One modality contains a spurious feature linearly correlated with the label at training strength β but uncorrelated at test. As β grows, the shortcut becomes increasingly attractive during training but provides no test-time signal. Figure 6 shows that vanilla fusion collapses to chance once $\beta \geq 0.6$, locking onto the shortcut and discarding the cross-modal signal. SynIB with oracle masking maintains $\sim$100% across all β, confirming that isolating synergistic information suffices. Without oracle annotations, both random and learned masks substantially improve robustness at every β, with the learned mask consistently closer to the oracle by identifying the features the unimodal head exploits and corrupting their complement.

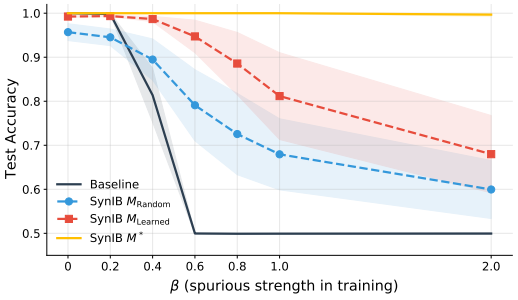


Figure 6: Test accuracy vs. spurious correlation strength β on bimodal XOR. Vanilla fusion collapses to chance as the shortcut strengthens. SynIB with oracle masking stays at $\sim$100%, while learned and random masks degrade gracefully, with the learned mask consistently closer to the oracle.

Coverage across PID compositions. Each example is generated from exactly one active source (unique, redundant, or synergistic), enabling controlled interpolation between unimodal-dominated and synergy-dominated regimes. Figure 7 reports performance across the information simplex. Without SynIB (Fig. 7a), accuracy degrades sharply as synergistic structure becomes dominant; with oracle masking (7b) it remains uniformly high. Random masking (7c) yields inconsistent gains, while

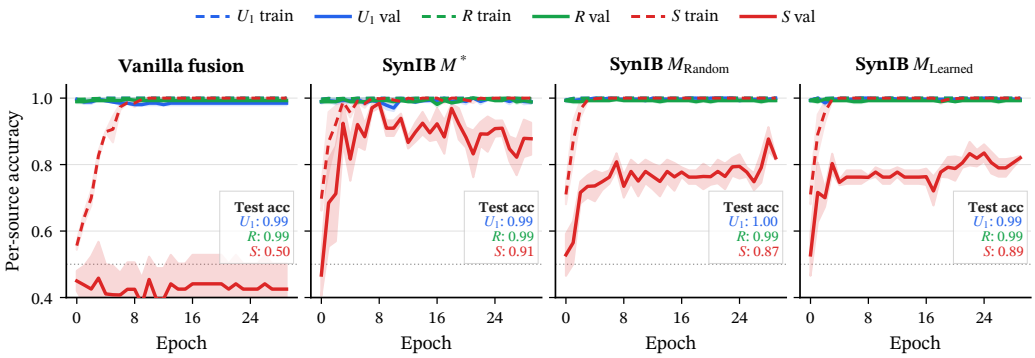


Figure 8: **PID-XOR training dynamics across all four methods.** Per-source training (solid) and validation (dashed) accuracy across 30 epochs, mean ± standard error over three seeds. Sources are colored by type: unique-to-modality-1 (U_1, blue), redundant (R, green), and synergistic (S, red). Final synergy test accuracies (annotated per panel): 0.50 vanilla, 0.91 oracle, 0.87 random, 0.89 learned. PID mixture $(p_{U_1}, p_{U_2}, p_R, p_S) = (0.45, 0, 0.45, 0.10)$. Vanilla fusion (left) fits U_1 and R on both splits, but synergy validation stalls at chance while training accuracy climbs, the signature of memorization on the scarce synergistic examples. The three SynIB variants ($M^\star$, M_{Random}, M_{Learned}) close this gap, lifting synergy validation accuracy without degrading U_1 or R.

the learned mask (7d) closely matches the oracle even when synergy is scarce, indicating that SynIB recovers synergistic structure from data without requiring annotations.

Per-source training dynamics. Figure 8 disaggregates accuracy by PID source across training, directly testing the diagnosis from Sec. 3.1: that synergy fails through overfitting on a scarce subset rather than through gradient interference or capacity limits. Vanilla fusion fits U_1 and R on both splits, but synergy shows the signature train–validation divergence, with training accuracy climbing while validation stalls at chance. The three SynIB variants close this gap: synergy train and validation track each other throughout training, with learned masking (0.89) tracking the oracle (0.91) within two points and random masking (0.87) close behind.

5 Conclusion

Multimodal models often default to unimodal shortcuts during training, even when the task rewards combining modalities. We argue this is an optimization problem: synergistic cues are scarce, so the model fits dominant unimodal signals first and never learns cross-modal interaction. SynIB, a PID-inspired training objective, addresses this by penalizing the model when its predictions remain confident under modality corruption, pushing it toward signals that emerge only from joint use of both modalities. Across synthetic tasks and six real-world datasets, including the controllable irony construction of CREMA-D and the Hateful Memes with pretrained backbones, SynIB consistently improves accuracy on examples requiring cross-modal reasoning while remaining competitive on the full test set. When synergy is scarce, the size of the validation synergy subset indicates how much SynIB can add, and selecting λ on validation keeps it from costing accuracy; when one modality dominates the task, the unmasked modality's prediction is a principled alternative to the uniform reference r (App. C.3). See App. A for limitations.

Broader Impact. Synergistic information underlies many of the settings where multimodal learning is most consequential, and methods that reliably train models to use cross-modal interactions, rather than defaulting to whichever modality is easiest to fit, expand the set of problems where multimodal learning can deliver on its premise. Better synergy does not, on its own, make a model trustworthy: a model that integrates modalities more effectively can still inherit biases from its training data, fail under distribution shift, or be applied in settings where higher accuracy is not the primary concern. We see synergy-targeted training as one ingredient in responsible multimodal learning, useful where cross-modal reasoning is the bottleneck and best paired with the auditing practices appropriate to each application.

Acknowledgments

This work was supported by KU Leuven Internal Funds under projects C3/24/023 (NEONAID: Neonatal Online Neuro-Analysis for Interpretation and Decision-support, bringing automated neonatal monitoring to the bedside) and C2M/25/034 (Inspire); by the Research Foundation - Flanders (FWO) under research project G046925N (Task- and device-agnostic Artificial Intelligence (AI) for EEG analysis); by the Methusalem programme under project METH/26/003 (Methusalem-BioMedAI – Explainable and generative AI for accelerating biomedical discoveries: from genome to function); and by Flanders Innovation & Entrepreneurship (VLAIO) under research project HBC.2025.0582 (MediSynth). This work was also supported by the MIT-Belgium KU Leuven Seed Fund, part of the MIT Global Seed Funds[2], through the MIT-Belgium program[3]. This work was further supported by the Innovative Health Initiative Joint Undertaking (IHI JU) under Grant Agreement No. 101252995 (HORIZON-JU-IHI-2025-09: GENz-trials, Revolutionizing Clinical Trials to Optimize Assessment of Healthcare Innovations), and by the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101095384 (HORIZON-HLTH-2022-IND-13: PHASE IV AI, Privacy compliant health data as a service for AI development). This research received funding from the Flemish Government (AI Research Program) and the Research Foundation - Flanders (FWO) through project number G0G2921N. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center).

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A. González. Gated multimodal units for information fusion. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pages 1298–1312. PMLR, 2022.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014.

Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 5(4):377–390, 2014.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629. Association for Computational Linguistics, 2019.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Yunfeng Fan, Wenchao Xu, Haozhao Wang, Junxiao Wang, and Song Guo. Pmr: Prototypical modal rebalance for multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20029–20038, 2023.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. *Advances in Neural Information Processing Systems*, 32, 2019.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Virgil Griffith and Christof Koch. Quantifying synergistic mutual information. In *Guided self-organization: inception*, pages 159–190. Springer, 2014.

[2] `https://cis.mit.edu/opportunities/seed-funding/available-funds/gsf-belgium`

[3] `https://cis.mit.edu/regions/europe/mit-belgium`

Md Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2046–2056. Association for Computational Linguistics, 2019.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979.

Cong Hua, Qianqian Xu, Shilong Bao, Zhiyong Yang, and Qingming Huang. Reconboost: boosting can achieve modality reconcilement. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yu Huang, Junyang Lin, Chang Zhou, Hongxia Yang, and Longbo Huang. Modality competition: What makes joint training of multi-modal network fail in deep learning?(provably). In *International Conference on Machine Learning*, pages 9226–9259. PMLR, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Baijun Ji, Tong Zhang, Yicheng Zou, Bojie Hu, and Si Shen. Increasing visual awareness in multimodal neural machine translation from an information theoretic perspective. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6755–6764, 2022.

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. *Advances in neural information processing systems*, 33:2611–2624, 2020.

Konstantinos Kontras, Christos Chatzichristos, Matthew Blaschko, and Maarten De Vos. Improving multimodal learning with multi-loss gradient modulation. *arXiv preprint arXiv:2405.07930*, 2024.

Konstantinos Kontras, Thomas Strypsteen, Christos Chatzichristos, Paul Pu Liang, Matthew B Blaschko, and Maarten De Vos. Balancing multimodal training through game-theoretic regularization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Hong Li, Xingyu Li, Pengbo Hu, Yinuo Lei, Chunxiao Li, and Yi Zhou. Boosting multi-modal model performance with adaptive gradient modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22214–22224, 2023.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. Multibench: Multiscale benchmarks for multimodal representation learning. *Advances in neural information processing systems*, 2021(DB1):1, 2021.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Advances in Neural Information Processing Systems*, 36:27351–27393, 2023.

Paul Pu Liang, Zihao Deng, Martin Q Ma, James Y Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized contrastive learning: Going beyond multi-view redundancy. *Advances in Neural Information Processing Systems*, 36, 2024a.

Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. *ACM Computing Surveys*, 56(10):1–42, 2024b.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8238–8247, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15638–15650, 2022.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

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 the conference. Association for computational linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Centralnet: a multilayer approach for multimodal fusion. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

Yake Wei and Di Hu. Mmpareto: boosting multimodal learning with innocent unimodal assistance. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yake Wei, Siwei Li, Ruoxuan Feng, and Di Hu. Diagnosing and re-learning for balanced multimodal learning. In *European Conference on Computer Vision*, pages 71–86. Springer, 2024.

Liangjian Wen, Qun Dai, Jianzhuang Liu, Jiangtao Zheng, Yong Dai, Dongkai Wang, Zhao Kang, Jun Wang, Zenglin Xu, and Jiang Duan. Infmasking: Unleashing synergistic information by contrastive multimodal interactions. *Advances in Neural Information Processing Systems*, 38:15529–15555, 2025.

Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

Jiayi Xin, Sukwon Yun, Jie Peng, Inyoung Choi, Jenna L Ballard, Tianlong Chen, and Qi Long. I2moe: Interpretable multimodal interaction-aware mixture-of-experts. *arXiv preprint arXiv:2505.19190*, 2025.

Yiqun Yao and Rada Mihalcea. Modality-specific learning rates for effective multimodal additive late-fusion. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1824–1834. Association for Computational Linguistics, 2022.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*, 2017.

Supplementary Material

A Limitations

We discuss four limitations of SynIB that we believe are important for interpreting our results and identifying directions for future work. Limitations of the NTK-based motivating analysis in Sec. 3.1 are discussed separately in App. K.

Dependence on the latent representation SynIB operates on representations produced by modality-specific encoders, and its effectiveness ultimately depends on whether synergistic features are expressible in the latent space those encoders produce. When the encoders strip out cross-modal structure during their forward pass, no loss-level objective can recover it. Self-supervised pretrained encoders such as CLIP partially mitigate this by preserving richer per-modality features, and our Hateful Memes experiments suggest that SynIB transfers cleanly to this setting. A more systematic study of how pretraining choices affect the synergy a fusion model can ultimately learn remains open.

Sensitivity of the learned mask The adversarial mask M_ψ is trained jointly with the SynIB objective and depends on hyperparameters governing sparsity (λ_M), optimization dynamics, and the unimodal predictor used to define the adversary. In our experiments these settings transferred reasonably across datasets without per-task tuning, but in regimes with very small synergy subsets, mask training can become unstable and may collapse to either trivial (no features kept) or near-identity (all features kept) solutions. In our sweeps (App. I.3), the only such case was near-identity: with 50 inner steps and a large λ_M, 1–2% of dimensions were masked on MUStARD and MOSI, and accuracy was unaffected. Random masking, while less precise, is more robust in these cases and provides a useful fallback.

Variational surrogate rather than exact decomposition Our objective is motivated by partial information decomposition, but we optimize a tractable variational surrogate of the Mutual Information Perturbed Difference rather than computing PID quantities directly. The surrogate coincides with MIPD when the variational family contains the true predictive distributions, and tracks the same direction in the loss landscape more generally; in particular, it is not guaranteed to be a one-sided bound on MIPD. SynIB therefore pushes the model toward predictions that depend on cross-modal interaction, but does not produce a formal decomposition of $I(Y; X_1, X_2)$ into unique, redundant, and synergistic components. The surrogate is sufficient for shaping training but does not, on its own, yield an explicit attribution of the learned model's predictions to unique, redundant, and synergistic components; that attribution is a separate analytical question we leave to future work.

Training-time overhead Each SynIB training step requires forward passes through the fusion model on both the intact inputs and one or more counterfactual inputs (one per masked subset of modalities, Sec. 3.6), increasing per-step compute by roughly a factor of $1+|\mathcal{A}|$: $1+n$ when masking single modalities, and 2^n-1 when masking all proper subsets, our recommended default for $n \geq 3$ (seven forward passes at n=3). The learned-masking variant additionally trains the mask network jointly. In our experiments this overhead was tolerable, and it remains practical for the modality counts of current benchmarks; for very large backbones or many modalities, sampling a few subsets to mask per step would keep the overhead fixed, which we have not tested and leave to future work.

A.1 NTK Analysis with SynIB

Section 3.1 used the empirical NTK to diagnose why synergistic examples fail to generalize under vanilla training: the largest per-source learning signal λ_g is allocated to synergy and no destructive interference appears between PID sources, yet synergistic validation loss rises while training loss drops. This appendix repeats the same diagnostics on a model trained with SynIB, verifying that the objective shifts these quantities in the direction the motivating analysis prescribes.

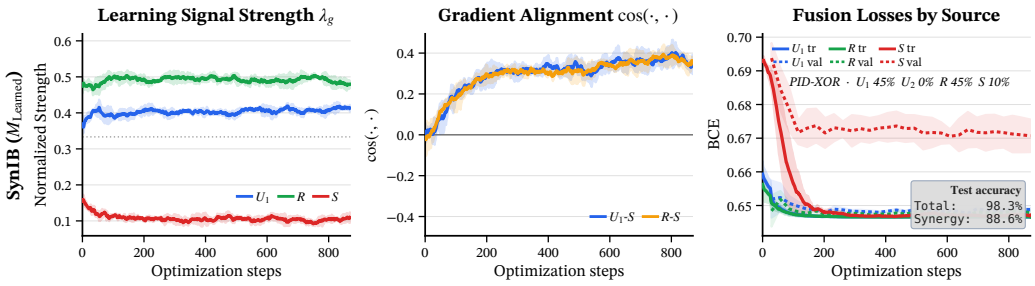


Figure 9: **Gradient geometry across PID sources under SynIB.** Training augments the vanilla setup with the SynIB learned-mask inner loop (λ=10); data, architecture, and optimiser are otherwise identical to Fig. 1. **Left:** per-source learning signal strength λ_g over training. SynIB redistributes mass toward U_1 and R, leaving synergy with the smallest share without starving it. **Center:** gradient alignment between source pairs stays near zero throughout training, ruling out destructive competition. **Right:** fusion BCE by source on intact inputs (solid train, dashed validation). Train and validation curves track closely on every source; the synergistic train–validation gap that opened under vanilla is now closed. Test accuracy: synergy 88.6%, total 98.3%.

Setup. We train on the PID-Controlled XOR data (App. F.3) at composition $(p_{U_1}, p_R, p_S) = (0.45, 0.45, 0.10)$, identical to the vanilla run in Fig. 1, and add SynIB with the learned-mask inner loop ($\lambda = 10$). Architecture, optimiser, batching, and seeds are unchanged.

Findings. Three changes appear in Fig. 9, each consistent with the SynIB design. First, the per-source learning signal redistributes: λ_{U_1} and λ_R rise while λ_S falls (left), so SynIB de-emphasizes the scarce synergistic examples that were being memorized under vanilla without starving any source. Second, gradient alignment between sources remains near zero throughout training (center), confirming that the redistribution does not introduce destructive competition. Third, and most importantly, the train–validation gap on the synergistic source closes (right): BCE on synergistic train and validation now track each other, in contrast to the diverging curves under vanilla. Combined with test accuracy of 88.6% on the synergy slice (98.3% overall), these findings indicate that SynIB reshapes the per-source signal in the way the motivating analysis prescribes, rather than simply injecting noise into training.

B Variational Surrogate for MIPD

B.1 Derivation of the SynIB Surrogate

This appendix derives the SynIB training objective as a tractable surrogate for $\mathrm{MIPD}_1 = H(Y \mid \tilde{X}_1, X_2) - H(Y \mid X_1, X_2)$. The two conditional entropies in this expression are not directly computable for high-dimensional inputs, since they require access to the true predictive distributions $p(y \mid x_1, x_2)$ and $p(y \mid \tilde{x}_1, x_2)$. We approximate each term using the variational distribution $q_\theta(y \mid \cdot)$ produced by the model, with $r(y)$ denoting a fixed reference distribution over $\mathcal{Y}$ chosen independently of the inputs and of the model parameters.

Upper bound on $H(Y \mid X_1, X_2)$ via cross-entropy. For the entropy under intact inputs, the standard variational upper bound due to Barber and Agakov [2004] applies. For any variational distribution $q_\theta(y \mid x_1, x_2)$,

$$\mathbb{E}_{p(x_1,x_2,y)}[-\log q_\theta(y \mid x_1, x_2)] = H(Y \mid X_1, X_2) + \mathbb{E}_{p(x_1,x_2)}\, D_{\mathrm{KL}}(p(\cdot \mid x_1, x_2) \,\|\, q_\theta(\cdot \mid x_1, x_2)), \tag{7}$$

which follows directly from expanding the cross-entropy and identifying the KL divergence between the true and variational conditionals. Since $D_{\mathrm{KL}} \geq 0$, this yields

$$H(Y \mid X_1, X_2) \leq \mathbb{E}_{p(x_1,x_2,y)}[-\log q_\theta(y \mid x_1, x_2)], \tag{8}$$

with equality if and only if $q_\theta(\cdot \mid x_1, x_2) = p(\cdot \mid x_1, x_2)$ almost everywhere. The bound's tightness is governed by the variational approximation error $\mathbb{E}_{p(x_1,x_2)} D_{\mathrm{KL}}(p \,\|\, q_\theta)$, which the cross-entropy training objective itself minimizes.

Reference-based identity for $H(Y \mid \tilde{X}_1, X_2)$. For the entropy under modality corruption, we use an exact identity that re-expresses conditional entropy in terms of a divergence to the reference r. Starting from the definition of conditional entropy and adding and subtracting $\log r(y)$,

$$\begin{aligned} H(Y \mid \tilde{X}_1, X_2) &= -\mathbb{E}_{p(\tilde{x}_1, x_2, y)}[\log p(y \mid \tilde{x}_1, x_2)] \\ &= -\mathbb{E}_{p(\tilde{x}_1, x_2, y)}\left[\log \frac{p(y \mid \tilde{x}_1, x_2)}{r(y)}\right] - \mathbb{E}_{p(y)}[\log r(y)] \\ &= C_r - \mathbb{E}_{p(\tilde{x}_1, x_2)}\, D_{\mathrm{KL}}(p(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot))\,, \end{aligned} \tag{9}$$

where $C_r := \mathbb{E}_{p(y)}[-\log r(y)]$ is the cross-entropy of the marginal $p(y)$ relative to r and is constant with respect to θ. This identity, used in the Information Bottleneck framework [Tishby et al., 2000, Alemi et al., 2016], expresses the conditional entropy as a constant minus the average divergence between the true conditional and the reference: when $p(\cdot \mid \tilde{x}_1, x_2)$ is far from r, the conditional entropy is small (predictions are sharp), and conversely.

Variational surrogate. Eq. (9) is exact but not directly usable, since the KL divergence involves the unknown $p(\cdot \mid \tilde{x}_1, x_2)$. We obtain a tractable expression by replacing the true conditional with the variational $q_\theta(\cdot \mid \tilde{x}_1, x_2)$:

$$\widehat{H}_\theta(Y \mid \tilde{X}_1, X_2) := C_r - \mathbb{E}_{p(\tilde{x}_1, x_2)}\, D_{\mathrm{KL}}(q_\theta(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot))\,. \tag{10}$$

The substitution defines a surrogate for $H(Y \mid \tilde{X}_1, X_2)$ that is computable from the model's predictions on samples drawn from $p(\tilde{x}_1, x_2)$. By construction, $\widehat{H}_\theta(Y \mid \tilde{X}_1, X_2) = H(Y \mid \tilde{X}_1, X_2)$ exactly when $q_\theta(\cdot \mid \tilde{x}_1, x_2) = p(\cdot \mid \tilde{x}_1, x_2)$.

Combining the two terms. Substituting Eq. (8) for $H(Y \mid X_1, X_2)$ and Eq. (10) for $H(Y \mid \tilde{X}_1, X_2)$ in MIPD_1 yields the SynIB surrogate

$$\begin{aligned} \widehat{\mathrm{MIPD}}_1 &= \widehat{H}_\theta(Y \mid \tilde{X}_1, X_2) - \mathbb{E}_{p(x_1, x_2, y)}[-\log q_\theta(y \mid x_1, x_2)] \\ &= C_r - \mathbb{E}_{p(\tilde{x}_1, x_2)}\, D_{\mathrm{KL}}(q_\theta(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot)) - \mathbb{E}_{p(x_1, x_2, y)}[-\log q_\theta(y \mid x_1, x_2)]. \end{aligned} \tag{11}$$

Up to the additive constant C_r, maximizing $\widehat{\mathrm{MIPD}}_1$ is equivalent to minimizing the SynIB loss: jointly minimizing the cross-entropy on intact inputs and the KL-to-reference on corrupted inputs.

Properties of the surrogate. Three observations clarify how $\widehat{\mathrm{MIPD}}_1$ relates to the true MIPD_1.

First, $\widehat{\mathrm{MIPD}}_1 = \mathrm{MIPD}_1$ whenever $q_\theta(\cdot \mid x_1, x_2) = p(\cdot \mid x_1, x_2)$ and $q_\theta(\cdot \mid \tilde{x}_1, x_2) = p(\cdot \mid \tilde{x}_1, x_2)$. In this case both the upper bound in Eq. (8) and the surrogate substitution in Eq. (10) are tight.

Second, the cross-entropy term in $\widehat{\mathrm{MIPD}}_1$ is itself the optimization signal that drives $q_\theta(\cdot \mid x_1, x_2)$ toward $p(\cdot \mid x_1, x_2)$, so the surrogate is self-correcting on the intact-input branch: improvements in q_θ tighten the bound in Eq. (8).

Third, the surrogate substitution in Eq. (10) is not a one-sided bound: the gap $\widehat{H}_\theta(Y \mid \tilde{X}_1, X_2) - H(Y \mid \tilde{X}_1, X_2) = \mathbb{E}_{p(\tilde{x}_1, x_2)}[D_{\mathrm{KL}}(p \,\|\, r) - D_{\mathrm{KL}}(q_\theta \,\|\, r)]$ can take either sign depending on the relative position of q_θ and p with respect to r. The surrogate should therefore be understood as a tractable target whose sign-correct alignment with MIPD_1 relies on q_θ approximating p in distribution, which the cross-entropy term promotes during training.

B.2 MIPD as a Proxy for CMI

Beyond serving as the optimization target, MIPD_1 admits a clean information-theoretic interpretation as a lower-bounded proxy function for the conditional mutual information $I(X_1; Y \mid X_2)$. Expanding the definition of MIPD in terms of conditional mutual informations gives $\mathrm{MIPD}_1 = I(X_1; Y \mid X_2) - I(\tilde{X}_1; Y \mid X_2)$, and since $I(\tilde{X}_1; Y \mid X_2) \geq 0$ for any random variable $\tilde{X}_1$, it follows directly that $\mathrm{MIPD}_1 \leq I(X_1; Y \mid X_2)$, with the gap given exactly by the residual conditional information retained after perturbation:

$$I(X_1; Y \mid X_2) - \mathrm{MIPD}_1 = I(\tilde{X}_1; Y \mid X_2). \tag{12}$$

This identity has a clear interpretation: MIPD recovers the full CMI exactly when the perturbation $\tilde{X}_1$ is conditionally uninformative about Y given X_2, i.e., when $Y \perp\!\!\!\perp \tilde{X}_1 \mid X_2$. In this case, all task-relevant information that X_1 contributed beyond X_2 has been removed by the corruption, and the drop in predictability MIPD_1 equals the conditional information X_1 originally carried. More generally, if the corruption leaves residual information $I(\tilde{X}_1; Y \mid X_2) \leq \varepsilon$, then MIPD approximates CMI to within ε.

The proxy quality thus depends entirely on how thoroughly the perturbation removes task-relevant information from X_1 beyond what X_2 already provides. Designing $\tilde{X}_1$ to make this residual small motivates the masking constructions in Sec. 3.5: random masking corrupts coordinates uniformly, while learned masking targets the features the unimodal predictor relies on, both aiming to drive $I(\tilde{X}_1; Y \mid X_2)$ as close to zero as possible while preserving the rest of the distribution.

C Variational Families and Closed-Form KL Divergences

The SynIB objective requires evaluating $-\log q_\theta(y \mid x_1, x_2)$ for the cross-entropy term and $D_{\mathrm{KL}}(q_\theta(\cdot \mid \tilde{x}_1, x_2) \,\|\, r(\cdot))$ for the KL term. Both quantities admit closed-form expressions for standard variational families, removing the need for stochastic estimators. We use the same family for both intact and corrupted inputs, with the conditioning input being the only difference between the two passes.

C.1 Gaussian Variational Family

For continuous targets $Y \in \mathbb{R}^d$, we model the variational predictive distribution as $q_\theta(y \mid \tilde{x}_1, x_2) = \mathcal{N}(y \mid \mu_\theta, \Sigma_\theta)$, with reference distribution $r(y) = \mathcal{N}(0, I)$. The mean μ_θ and covariance Σ_θ are outputs of the variational decoder and depend on $(\tilde{x}_1, x_2)$. The KL divergence admits the closed form

$$D_{\mathrm{KL}}(q_\theta \,\|\, r) = \tfrac{1}{2}\left(\|\mu_\theta\|_2^2 + \operatorname{tr}\Sigma_\theta - \log\det\Sigma_\theta - d\right), \tag{13}$$

which is minimized when $q_\theta = r$ (i.e., $\mu_\theta = 0$, $\Sigma_\theta = I$) and grows with both the magnitude of μ_θ and the deviation of Σ_θ from the identity, providing a differentiable uncertainty penalty under modality corruption.

C.2 Categorical Variational Family

For categorical targets $Y \in \{1, \ldots, K\}$, we model the variational predictive distribution as $q_\theta(y \mid \tilde{x}_1, x_2) = \mathrm{Categorical}(y; \pi_\theta)$, with reference $r(y) = \mathrm{Categorical}(y; \rho)$, where $\pi_\theta = \pi_\theta(\tilde{x}_1, x_2) \in \Delta^{K-1}$ is the output of the variational decoder and $\rho \in \Delta^{K-1}$ is a fixed reference distribution (we use $\rho_k = 1/K$, the uniform prior, in all experiments). The KL divergence admits the closed form

$$D_{\mathrm{KL}}(q_\theta \,\|\, r) = \sum_{k=1}^{K} \pi_{\theta,k} \log \frac{\pi_{\theta,k}}{\rho_k}. \tag{14}$$

The penalty is minimized when $\pi_\theta = \rho$ and grows as π_θ concentrates on any single class, discouraging confident predictions under modality corruption. The binary case $K = 2$ recovers the Bernoulli form $D_{\mathrm{KL}}(q_\theta \,\|\, r) = \pi_\theta \log(\pi_\theta/\rho) + (1 - \pi_\theta)\log((1 - \pi_\theta)/(1 - \rho))$, used for tasks with binary targets.

C.3 Choice of the reference distribution

The KL term of Eq. (6) pulls the prediction under corruption, $q_\theta(\cdot \mid \tilde{x}_1, x_2)$, toward r, so this prediction approximates the true conditional $p(\cdot \mid \tilde{x}_1, x_2)$ exactly to the extent that r does. With a uniform r, this holds where the method acts: if the corruption succeeds ($Y \perp \tilde{X}_1 \mid X_2$, the design criterion of Sec. 3.5), then $p(y \mid \tilde{x}_1, x_2) = p(y \mid x_2)$, which is near-uniform precisely on synergy-dependent examples, where X_2 alone is uninformative. Elsewhere the two differ, and the term acts as a deliberate penalty on unimodal reliance rather than as an estimator. A uniform r penalizes all confidence under corruption, including what X_2 legitimately supplies, and so also shifts weight from unique and redundant information toward synergy; the complementary unimodal prediction $q_{\theta_{c_2}}(\cdot \mid x_2)$ exempts exactly that part.

Table 2 compares three references: uniform ($1/K$), the empirical label prior $\hat{p}(y)$, and $q_{\theta_{c_2}}(\cdot \mid x_2)$. On four of the five benchmarks, the differences on the synergy metric are within fold-to-fold variation; on CREMA-D-Irony, the uniform reference gives clearly higher irony-class F1 (22.1 vs. 8.1 and 9.6). Whole-test accuracy is essentially the same for all three references. We therefore use a uniform r by default: it needs no auxiliary head, no extra forward pass, and no per-dataset tuning. The one case where we would deviate is a task known to be largely solvable from one modality, as judged from unimodal validation accuracy before training. There a uniform r also penalizes the confidence that modality legitimately supplies, and $q_{\theta_{c_2}}(\cdot \mid x_2)$ is the principled alternative, at the cost of keeping a unimodal head as the reference.

Table 2: Choice of the reference r: synergy-subset accuracy (irony-class F1 on CREMA-D-Irony at $\alpha = 0.5$), mean $\pm$ sd over 3 folds. All rows share one SynIB configuration per benchmark, which differs from that of the main experiments, so values are comparable within a column but not to Table 5 or Table 4. Whole-test accuracy is essentially unchanged across references.

Reference r	CREMA-D-Irony	Hateful Memes	MUStARD	MOSI	UR-Funny
Uniform ($1/K$)	22.1$\pm$7.9	48.2$\pm$12.2	33.9$\pm$4.3	17.4$\pm$3.3	13.9$\pm$6.8
Label prior $\hat{p}(y)$	8.1$\pm$5.5	44.0$\pm$4.5	32.1$\pm$2.6	15.5$\pm$3.2	11.8$\pm$0.7
Unimodal $q_{\theta_{c_2}}(\cdot \mid x_2)$	9.6$\pm$7.0	43.4$\pm$10.3	34.2$\pm$2.5	14.6$\pm$4.2	16.5$\pm$6.9

D Masking Operator Instantiations

This appendix specifies the concrete realizations of the masking operator $\mathcal{T}(\cdot; M)$ introduced in Sec. 3.5. All variants implement the same semantics, selectively corrupting a subset of representation coordinates while leaving the rest intact, and differ only in how the mask M is constructed and how the suppression is realized within a given architecture. Throughout this appendix we follow the convention from the main text: $M_j = 1$ denotes a corrupted coordinate, $M_j = 0$ a preserved one.

Feature replacement via gating. In non-attention-based fusion networks, masking is implemented by replacing corrupted coordinates with noise through the feature-wise gate

$$\tilde{x}_1 = (1 - M) \odot x_1 + M \odot \epsilon, \tag{15}$$

where $M \in \{0, 1\}^{d_1}$ and ϵ is sampled independently of the target Y. The construction preserves dimensionality and, by design, removes the task-relevant signal carried by the corrupted features while leaving the rest of the input distribution intact. The same expression applies when masking is performed in the latent space: substituting $z_1 = f_{\theta_{c_1}}(x_1)$ for x_1 yields the corresponding latent-space gate.

Oracle masking. In synthetic settings where the synergistic coordinates of X_1 are known by construction, we define an oracle mask

$$M^\star_j = \mathbf{1}[j \in S_{\text{syn}}], \tag{16}$$

which corrupts exactly the designated synergistic features and preserves everything else. Applied through Eq. (15), this isolates the contribution of synergy by removing only the coordinates whose information is recoverable from the joint (X_1, X_2) but not from either modality alone. Oracle masking is used exclusively for diagnostic experiments and serves as an upper-bound reference point: it characterizes how SynIB would behave if it could perfectly identify synergistic features, without claiming that this information is available in real-world settings.

Random masking. For stochastic masking, each coordinate is independently corrupted according to $M_j \sim \text{Bernoulli}(\pi)$, where $\pi \in (0, 1)$ controls the expected corruption level. Random masking makes no assumption about which coordinates carry which type of information, providing a task-agnostic baseline that removes signal uniformly across dimensions and surfaces synergy whenever any corrupted subset breaks the joint prediction.

To avoid introducing out-of-distribution artifacts when corrupting, the noise ϵ in Eq. (15) is sampled to match the marginal statistics of the representation being corrupted. Let $z_1 = f_{\theta_{c_1}}(x_1)$ and maintain

exponential moving averages over minibatches $\mathcal{B}$:

$$\begin{aligned}\mu_t &= \alpha\mu_{t-1} + (1-\alpha)\,\mathbb{E}_{\mathcal{B}}[z_1],\\ \sigma_t^2 &= \alpha\sigma_{t-1}^2 + (1-\alpha)\,\mathbb{E}_{\mathcal{B}}\left[(z_1 - \mathbb{E}_{\mathcal{B}}[z_1])^2\right].\end{aligned} \tag{17}$$

Noise is then sampled as $\epsilon \sim \mathcal{N}(\mu_t, \mathrm{diag}(\sigma_t^2))$. The momentum α controls the smoothness of the running statistics; we use $\alpha = 0.99$ in all experiments. To prevent the model from exploiting structure in the replacement distribution, the same noise process is applied to fully corrupted counterfactual inputs during training.

Learned masking. The learned mask M_{ψ_1} introduced in Sec. 3.5 flags the minimum set of coordinates whose corruption collapses unimodal prediction. Training M_{ψ_1} adversarially against the unimodal predictor $f_{\theta_{c_1}}$ via Eq. (5) therefore identifies the unimodally important features of X_1. The SynIB counterfactual then uses the complement $\bar{M}_{\psi_1} = \mathbf{1} - M_{\psi_1}$ as its corruption mask in Eq. (15), leaving the unimodally important coordinates intact and corrupting the rest. Confidence under this corruption signals reliance on cues that survive the removal of X_1's non-unimodal signal, which the SynIB KL term penalizes. The mask network M_{ψ_1} is parameterized as a small MLP producing logits $\ell \in \mathbb{R}^{d_1}$, with binary samples drawn through a Gumbel-softmax relaxation [Jang et al., 2016] to enable gradient flow during joint training.

E Likelihood Perspective on Learned Masking

Learned masking can be interpreted as a structured perturbation of the conditional likelihood implicitly defined by the unimodal predictor $f_{\theta_{c_1}}$. This appendix makes the connection explicit and clarifies how the learned mask differs from the fixed alternatives (random, oracle).

Cross-entropy training of $f_{\theta_{c_1}}$ implicitly defines a conditional likelihood $p_{\theta_{c_1}}(y \mid z_1)$ over the target given the modality-1 representation, with the cross-entropy loss equal to its negative log-likelihood under the data distribution:

$$\mathrm{CE}\big(f_{\theta_{c_1}}(z_1),\, y\big) = -\mathbb{E}_{p(y,z_1)}\left[\log p_{\theta_{c_1}}(y \mid z_1)\right]. \tag{18}$$

Perturbations of the representation that increase cross-entropy therefore correspond directly to degradations of this conditional log-likelihood.

The learned mask M_{ψ_1} exploits this relationship by seeking the sparsest perturbation that maximally degrades the likelihood. With $\tilde{Z}_1 = \mathcal{T}(Z_1;\, M_{\psi_1})$ denoting the gated representation under the new convention ($M_{\psi_1,j} = 1$ corrupts coordinate j, $M_{\psi_1,j} = 0$ preserves it), the mask training objective in Eq. (5) can be equivalently written as

$$\psi_1^\star = \arg\max_{\psi_1}\; \mathbb{E}_{p(y,z_1)}\left[\log \frac{p_{\theta_{c_1}}(y \mid Z_1)}{p_{\theta_{c_1}}(y \mid \tilde{Z}_1)}\right] - \lambda_M \|M_{\psi_1}\|_1, \tag{19}$$

where the first term measures the average log-likelihood drop induced by the perturbation and the second imposes an ℓ_1 relaxation of an ℓ_0 sparsity constraint on the corruption set. Solutions concentrate corruption on the minimal subset of representation coordinates whose removal most degrades the unimodal likelihood, identifying the features $f_{\theta_{c_1}}$ relies on for prediction.

This view places the three masking strategies on a common axis. Random masking ($M_j \sim \mathrm{Bernoulli}(\pi)$) corrupts coordinates uniformly without consulting the likelihood. Oracle masking corrupts a fixed set S_{syn} defined by ground-truth synergy structure available only in synthetic settings. Learned masking adaptively concentrates corruption on the coordinates that most reduce the likelihood of $f_{\theta_{c_1}}$, which we then complement in the SynIB counterfactual to leave only the unimodally-important features intact.

F Additional Synthetic Experiment Details

We use two synthetic XOR datasets, each targeting a different aspect of SynIB's behavior. Sec. F.1 (*Spurious XOR*) tests robustness when one modality contains a label-correlated shortcut whose strength sweeps continuously between absent and dominant. Sec. F.3 (*PID-Controlled XOR*) generates

examples whose label is determined by a single PID source per sample, allowing the dataset's mixture of unique, redundant, and synergistic information to be controlled directly. Both datasets use the modality-indexing convention from the main text (X_1, X_2) and the masking convention $M_j = 1$ corrupts, $M_j = 0$ preserves.

F.1 Spurious XOR

F.1.1 Task and Data Generation

The Spurious XOR task is a bimodal binary classification problem in which neither modality alone is informative about y, but their XOR is. Latent bits $b_1, b_2 \sim \mathrm{Bernoulli}(0.5)$ are sampled independently and the label is set to $y = b_1 \oplus b_2$, with modality X_m encoding bit b_m. Modality X_1 additionally contains a spurious feature whose correlation with y during training is controlled by a parameter β and is removed at test time, so a model that latches onto the shortcut fails to generalize.

Both modalities are 64-dimensional and partitioned into disjoint coordinate blocks. Modality 1 contains a 4-coordinate signal block at $[0, 4)$ encoding b_1, a 6-coordinate spurious block at $[4, 10)$ correlated with y at training time, and 54 pure-noise coordinates at $[10, 64)$. Modality 2 contains a 4-coordinate signal block at $[0, 4)$ encoding b_2 and 60 pure-noise coordinates at $[4, 64)$. Pure-noise coordinates are sampled as $\mathcal{N}(0, 1)$.

Both signal blocks are mean-shifted Gaussians, $X_m^{\mathrm{sig}} = \mu \cdot (2b_m - 1) \cdot \mathbf{1}_4 + \sigma_{\mathrm{sig}}\, \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_4)$, $\mu = 2.2$, and $\sigma_{\mathrm{sig}} = 0.55$. The spurious block in X_1 is a continuous mean-shifted Gaussian whose label-correlation strength is controlled by $\beta \geq 0$:

$$X_1^{\mathrm{spur}} = 1.3 \cdot \Big[\beta \cdot (2y_{\mathrm{eff}} - 1) \cdot \mathbf{1}_6 + \sqrt{\max(0, 1 - \beta^2)} \cdot \varepsilon \Big], \quad \varepsilon \sim \mathcal{N}(0, I_6), \tag{20}$$

with $y_{\mathrm{eff}} = y$ at training time and $y_{\mathrm{eff}} = 0$ at test time. The parameter β is a continuous SNR knob, not a probability: at $\beta = 0$ the spur is pure noise; at $\beta = 1$ the noise term vanishes and the spur becomes deterministic in y; the $\max(0, \cdot)$ clamp permits values $\beta > 1$, which further amplify the deterministic mean-shift while keeping the noise term zero. We sweep $\beta \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0, 2.0\}$. The training set contains 300 samples, the validation set 120, and the test set 30,000; the small training set is deliberate and gives the spurious shortcut a real chance to dominate the cross-modal XOR signal during fitting.

F.1.2 Model, Objective, and Mask Variants

The fusion network has modality-specific encoders, two unimodal heads, and a fusion head. Each encoder f_{θ_m} is a single hidden layer, $\mathrm{Linear}(64 \to 16) \to \mathrm{ReLU} \to \mathrm{Dropout}$. Unimodal heads are $\mathrm{Linear}(16 \to 1)$, trained against y with weight $\lambda_{\mathrm{uni}} = 0.05$. The fusion head concatenates the two encoder outputs and applies $\mathrm{Linear}(32 \to 32) \to \mathrm{ReLU} \to \mathrm{Dropout} \to \mathrm{Linear}(32 \to 1)$, producing a single logit trained with binary cross-entropy. SynIB regularizes the fusion logit symmetrically across modalities,

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \sum_{i \in \{1,2\}} D_{\mathrm{KL}}\Big(P_f(\cdot \mid \tilde{X}_i, X_{\neg i}) \,\Big\|\, \mathrm{Bernoulli}(0.5) \Big), \tag{21}$$

where $X_{\neg i}$ denotes the unmasked modality. The reference distribution is uniform over the binary outcomes; only the fusion logit is regularized, not the unimodal heads.

Corrupted coordinates are replaced via $\tilde{x} = (1 - M) \odot x + M \odot \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$. The oracle mask $M^\star$ flags the signal blocks $[0, 4)$ of both modalities (the coordinates encoding b_1, b_2); the spurious block in X_1 is not flagged, so any confident prediction under oracle corruption would have to rely on the shortcut. Random masking corrupts each coordinate of each modality independently with probability $\pi = 0.4$. Learned masking uses per-example, per-coordinate logits initialized at -4, with an inner loop of 50 Adam steps (lr $= 3.0$, sparsity $\lambda_M = 0.03$, temperature $\tau = 1.0$) finding the smallest mask whose corruption breaks the unimodal head; the discovered mask is hard-thresholded at 0.5, and the SynIB pass corrupts the complement, leaving the unimodally-important coordinates intact.

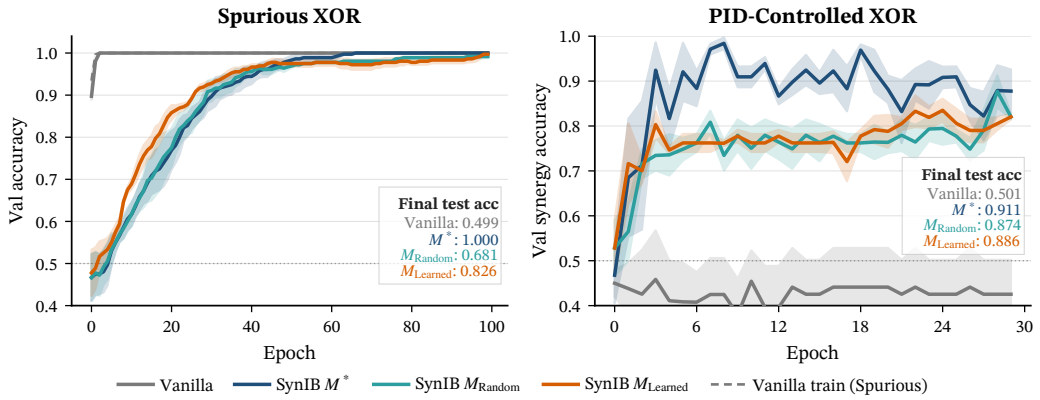


Figure 10: **Validation accuracy over training on both synthetic benchmarks.** Mean ± standard error across three seeds. **Left**: Spurious XOR ($\beta = 1.0$). All four methods reach ≈ 1.0 on the in-distribution validation set, fitting the training distribution equally well. The gray dashed curve shows training accuracy for vanilla fusion, which saturates within the first epoch as the model locks onto the shortcut. The annotated final out-of-distribution test accuracies reveal the divergence: vanilla collapses to chance (0.50) while oracle masking recovers fully (1.00); learned masking reaches 0.83 and random masking 0.68. Random masking shows the largest seed-to-seed variance (per-seed test accuracies 0.58–0.78), reflecting sensitivity to which coordinates the random mask happens to corrupt. **Right:** PID-Controlled XOR. Synergy-restricted validation accuracy at simplex point $(0.45, 0, 0.45, 0.10)$. Vanilla plateaus at chance on the synergy slice; oracle, learned, and random masking each rise above 0.85, with learned masking (0.89) tracking oracle (0.91) within three percentage points.

F.1.3 Training Configuration

We train with Adam, batch size 128, for 100 epochs, across seeds $\{0, 1, 2\}$ and dropout 0. The baseline uses learning rate 10^{-2} and weight decay 10^{-3}; the SynIB variants use learning rate 3×10^{-4} and weight decay 10^{-5}, with regularization strength $\lambda = 10$. Validation accuracy curves over training are reported in Figure 10 (left panel) with final test accuracies. Reported numbers are mean ± standard error across the 3 seeds.

F.2 Computational Overhead

We analyse the cost of SynIB by separating the per-sample cost of the modality encoders $f_{\theta_1}, f_{\theta_2}$ from that of the classification heads, i.e. the fusion network $f_{\theta_{c_{12}}}$ and the unimodal heads $f_{\theta_{c_1}}, f_{\theta_{c_2}}$. Let $\mathrm{cost}_{\mathrm{enc}}$ denote the combined encoder cost per sample (computing Z_1, Z_2) and cost_c the combined head cost per sample (the fusion and unimodal predictions on fixed Z_1, Z_2), with batch size N. A standard supervised step then costs $O\big(N(\mathrm{cost}_{\mathrm{enc}} + \mathrm{cost}_c)\big)$.

During training, SynIB adds to this supervised pass one counterfactual evaluation per modality, in which that modality is masked and the resulting prediction is penalised against the reference distribution. Because masking is applied to the encoder outputs Z_1, Z_2 rather than to the raw inputs, these counterfactual passes do not re-run the encoders f_{θ_i}; only the heads $f_{\theta_{c_{12}}}, f_{\theta_{c_i}}$ are recomputed on the masked representations. The two masking variants differ only in how the mask is produced (with S=20 inner steps in our experiments):

- M_{Random} samples the mask rather than optimising it, adding two head evaluations per step (one per masked modality), an extra $O\big(2\,N\,\mathrm{cost}_c\big)$, for a total of $O\big(N\,\mathrm{cost}_{\mathrm{enc}} + 3\,N\,\mathrm{cost}_c\big)$.
- M_{Learned} runs, for each masked modality, an S-step inner loop that optimises a lightweight per-feature gate (a vector the size of one Z_i). This gives $2(S{+}1)$ head evaluations per step, an extra $O\big(2(S{+}1)\,N\,\mathrm{cost}_c\big)$, for a total of $O\big(N\,\mathrm{cost}_{\mathrm{enc}} + (1 + 2(S{+}1))\,N\,\mathrm{cost}_c\big)$.

In both variants the additional computation touches only the heads, while the encoders are run once. In the common regime where each modality encoder is far larger than the heads ($\mathrm{cost}_{\mathrm{enc}} \gg \mathrm{cost}_c$),

and especially in our frozen-feature setup where Z_1, Z_2 are precomputed, this overhead is a small fraction of the supervised pass for both variants, even given the larger nominal pass count of M_{Learned}. The memory footprint follows the same pattern: a single additional masked activation set over the heads, plus the gate parameters for M_{Learned}, both negligible relative to the encoder activations.

At inference, SynIB has no cost at all. The masking, the optional inner loop, and the reference-distribution penalty are training-time components that are discarded once training ends. For both variants the deployed model is exactly the trained network $f_{\theta_{c_{12}}}(f_{\theta_1}(X_1), f_{\theta_2}(X_2))$, with the same parameters θ and the same number of FLOPs as the vanilla fusion baseline, so SynIB incurs zero inference overhead regardless of the masking variant.

F.3 PID-Controlled XOR

F.3.1 Task and Data Generation

The PID-Controlled XOR task is a bimodal binary classification problem in which each sample's label is generated by a single PID source: unique to modality 1 (U_1), unique to modality 2 (U_2), redundant (R), or synergistic (S). The dataset's mixture is set by a categorical distribution $p(A) = (p_{U_1}, p_{U_2}, p_R, p_S)$, allowing direct control over the prevalence of each information type. For Figure 7 we hold $p_{U_2} = 0$ and sweep the remaining three coordinates over the probability simplex on a regular grid with step 0.05.

Both modalities are 32-dimensional and partitioned into four disjoint blocks of 6, 6, 6, and 14 coordinates. The first block carries the modality's unique signal (U_1 for modality 1, U_2 for modality 2), the second block carries the redundant signal, the third block carries the synergistic signal, and the fourth block is pure noise. Block coordinate sets are fixed across samples.

Each sample is generated by initializing all coordinates of X_1, X_2 as i.i.d. $\mathcal{N}(0, 1)$, sampling $y \sim \text{Bernoulli}(0.5)$ independently, sampling an active source $A \sim p(A)$, overwriting the block coordinates of the active source with label-aligned signal as described below, and finally standardizing each coordinate using the training-set mean and standard deviation. Blocks not corresponding to the active source remain at their initial $\mathcal{N}(0, 1)$ noise.

Three pairs of i.i.d. random projection matrices $P_{u_m}, P_{r_m}, P_{s_m} \in \mathbb{R}^{d_m \times 4}$ are sampled once at script load (seeded for reproducibility) with entries i.i.d. $\mathcal{N}(0, 0.5^2)$. Sign-aligned latents are constructed by sampling $z \sim \mathcal{N}(0, s^2 I_4)$ for the appropriate source-specific scale s and applying $z \leftarrow (2b - 1) \cdot |z|$ elementwise for the relevant bit b. The full projected vector is computed and only the corresponding block coordinates are written back into the modality. We use $s_u = s_r = s_s = 3$.

When source U_m is active, z is sampled with $b = y$ and $X_m[\text{unique}] = (P_{u_m} z)[\text{unique}]$; the other modality's unique block remains noise. When source R is active, a single shared latent z_R is sign-aligned with y and projected through different matrices P_{r_1}, P_{r_2} into the redundant blocks of both modalities, so the two redundant blocks are correlated through z_R but not identical. When source S is active, a latent bit $b \sim \text{Bernoulli}(0.5)$ is sampled with $b_1 = b$ and $b_2 = b \oplus y$ so that $b_1 \oplus b_2 = y$; independent latents $z^{(m)}$ are sign-aligned with b_m and projected into modality m's synergistic block, making each modality alone uninformative about y while their XOR recovers it. The training set contains 1800 samples per simplex point, with 200 held out for validation and 4000 for test.

F.3.2 Model, Objective, and Mask Variants

The PID-XOR model uses substantially more capacity than the Spurious-XOR model, reflecting the harder task of distinguishing four signal types across more coordinates. Each encoder is two hidden layers, $\text{Linear}(32 \to 1024) \to \text{ReLU} \to \text{Dropout} \to \text{Linear}(1024 \to 1024) \to \text{ReLU}$. Unimodal heads are $\text{Linear}(1024 \to 1024) \to \text{ReLU} \to \text{Linear}(1024 \to 1)$, with the unimodal-head loss entering the total objective with weight $\lambda_{\text{uni}} = 1.0$. The fusion head concatenates the two encoder outputs (2048-d) and applies $\text{Linear}(2048 \to 128) \to \text{ReLU} \to \text{Dropout} \to \text{Linear}(128 \to 128) \to \text{ReLU} \to \text{Linear}(128 \to 1)$. The SynIB objective takes the same form as in Sec. F.1, with $\text{Bernoulli}(0.5)$ as the reference and KL applied to the fusion logit.

The same gating mechanism is used, with replacement noise $\mathcal{N}(0, 1)$ per coordinate. The oracle mask $M^\star$ corrupts only the synergistic blocks of both modalities; unique, redundant, and noise blocks are preserved. Random masking corrupts each coordinate of each modality independently

with probability $\pi = 0.5$. Learned masking uses an inner loop of 20 Adam steps (lr $= 0.1$, $\tau = 1.0$); the hard binary mask discovered by the inner loop is used as the corruption pattern in the outer SynIB pass.

F.3.3 Sources spread across all coordinates

Learned masking (Sec. 3.5) flags, per example, the coordinates a given head relies on for its prediction. This is a measured property of the trained predictor, not a labelling of coordinates by information type: no coordinate is assumed to carry only unique or only synergistic information, and the gate of Eq. (15) can corrupt any subset. The assumption is weaker, namely that the unimodal and fusion heads rely on a sparse set of coordinates. In the PID-XOR data above, however, each source occupies its own block of coordinates, which could favour coordinate-wise masking. To remove this structure, we mix each modality's 32 coordinates after data generation with a fixed invertible map that spreads every source over all coordinates, at two severities: a saturating tanh applied to a random rotation, and a frozen saturating two-layer MLP. Everything else follows App. F.3.2.

Two checks validate the setup. The task survives the mixing: vanilla fusion still exceeds 0.98 accuracy on unique and redundant examples while staying at chance on synergistic ones. And the coordinate structure is gone: without mixing, a source becomes unrecoverable once 16% of a modality's coordinates are corrupted, consistent with the six-coordinate blocks, whereas after the two maps this takes 81% and 87% (Table 3). No small subset of coordinates carries a source, so any effective mask must be dense.

Under both maps, SynIB still classifies synergistic examples well above chance while vanilla fusion stays at chance (Table 3): random masking reaches 0.834 and 0.813, and learned masking 0.874 and 0.769, against 0.876 and 0.862 without mixing. Learned masking is the more affected under the frozen MLP, but in both cases the gains do not depend on unique and synergistic information occupying separate coordinates.

Table 3: Synergy-slice test accuracy on PID-XOR when each modality's coordinates are mixed by a fixed invertible map; mean $\pm$ standard error over 3 seeds, chance ≈ 0.50. The identity row is re-run with the mixed variants. Source spread: fraction of a modality's 32 coordinates that must be corrupted before a source becomes unrecoverable, averaged over the four sources.

Mixing map	Source spread	Vanilla	SynIB M_{Random}	SynIB M_{Learned}
None (identity)	16%	0.504$\pm$0.011	0.876$\pm$0.011	0.862$\pm$0.013
tanh $\circ$ random rotation	81%	0.488$\pm$0.018	0.834$\pm$0.014	0.874$\pm$0.010
Frozen saturating MLP	87%	0.498$\pm$0.011	0.813$\pm$0.007	0.769$\pm$0.025

G Disentangling Masking Effects from SynIB

An arising question is whether the gains from SynIB stem from the proposed information-theoretic objective or simply from exposing the model to masked inputs during training. To rule out the latter, we compare SynIB against a masking-only baseline that uses the same mask-construction mechanism as SynIB but applies it as input augmentation rather than as a counterfactual KL penalty: each batch is augmented with masked copies of the original samples, and training proceeds with cross-entropy on the union of intact and masked inputs.

We instantiate two variants of this baseline that mirror SynIB's mask choices: *masked input with oracle* corrupts the unique and redundant blocks (preserving the synergistic block) so the model is exposed to inputs that carry only synergistic signal, and *masked input with random masking* applies a random Bernoulli mask matching the SynIB-random configuration.

Figure 11 reports the comparison across the PID simplex. The masking-only baselines yield limited improvements over no regularization and fail to recover the synergistic structure, particularly when synergy is dominant. SynIB, in contrast, achieves substantially higher accuracy across the simplex and on the synergy-restricted subset. The gap indicates that input augmentation alone is insufficient: SynIB's gains depend on explicitly penalizing confident predictions under modality corruption, not on data exposure to masked inputs.

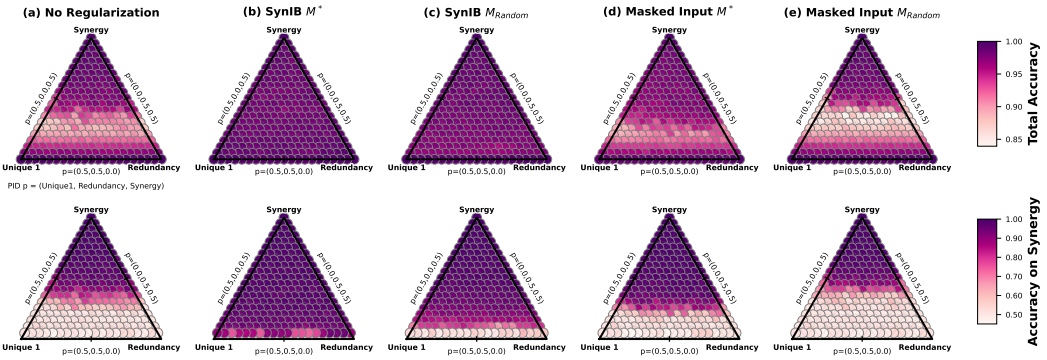


Figure 11: Masking-based baselines vs SynIB across PID compositions. Columns: (a) no regularization, (b) SynIB with oracle masking $M^\star$, (c) SynIB with random masking, (d) masked-input training with oracle, (e) masked-input training with random masking. Triangles sweep the PID simplex over (p_{U_1}, p_R, p_S). Top row: total accuracy; bottom row: accuracy restricted to synergistic samples. Masked-input baselines yield limited gains, while SynIB recovers synergistic structure across the simplex.

H Synergy-Driven Irony Dataset Construction

This appendix provides the construction details for the synthetic irony benchmark introduced in Sec. 4.1, along with the corresponding model architecture, training setup, and evaluation protocol.

H.1 Base Dataset and Splits

We start from CREMA-D [Cao et al., 2014], a crowd-sourced audio–visual emotion recognition dataset of short clips in which actors speak sentences with a target emotion. Following the standard 6-class taxonomy, we use neutral (NEU), happy (HAP), sad (SAD), fearful (FEA), disgusted (DIS), and angry (ANG); we add a synthetic 7th class for irony. We use a 5-fold speaker-disjoint split with no actor appearing in more than one of (train, val, test) within any fold. After dropping samples missing audio, video, or face crops, each fold contains 5,820 samples (approximately 4,043 train / 878 val / 899 test); base classes are mildly imbalanced with HAP largest and SAD smallest at a roughly $1.5\times$ ratio.

H.2 Mutation Procedure

Let $D_{\text{original}} = \{(V_i, A_i, L_i)\}_{i=1}^{N}$ denote the original set of (video, audio, label) triplets, and let $\bar{N}$ denote the mean number of samples per base class within a given split. The irony rate α controls the number of mutated samples, $|I| = \lfloor \alpha \bar{N} \rfloor$, drawn uniformly without replacement from the base-class pool of that split. For each selected index i with base label L_i, we sample a donor index j from a designated set of contradiction labels and form a mutated triplet $(V_i, A_j, L_{\text{irony}})$. The original audio is replaced by the donor's audio while video and identity are preserved; the new label is the irony class. The resulting dataset is

$$D_{\text{new}} = \{(V_i, A_j, L_{\text{irony}})\}_{i \in I} \cup \{(V_i, A_i, L_i)\}_{i \notin I}. \tag{22}$$

Mutation is in-place: each mutated sample replaces its original, so the total dataset size is preserved and the irony class population is controlled solely by α. We sweep $\alpha \in \{0.1, 0.3, 0.5, 0.8, 1.0, 2.0\}$.

The donor set for each base class is fixed by a semantic contradiction map: HAP is paired with one of $\{$SAD, FEA, DIS, ANG$\}$, while SAD, FEA, DIS, and ANG are each paired with HAP. NEU samples have no semantically opposite class in this taxonomy and are paired with any non-matching label uniformly at random. This map ensures that ironic pairings produce strong cross-modal contradictions rather than subtle disagreements between similar emotions. Donors are drawn from the same split as the mutated sample (train donors for train mutations, etc.) to prevent identity leakage across train and test, and the same α is applied to all three splits. Each mutated sample is, by construction, individually consistent with both its visual base emotion and its donor audio's emotion: neither modality alone

identifies the sample as ironic, so the irony label is recoverable only from the contradiction between modalities.

H.3 Feature Representations

Audio is represented as a log-magnitude short-time Fourier transform (STFT) spectrogram. Waveforms are resampled to 22,050 Hz, tiled and cropped to 3 seconds, and clipped to $[-1, 1]$ before computing an STFT with $n_{\text{FFT}} = 512$ and hop length 353. The resulting spectrogram is $\log(|\cdot| + 10^{-7})$, producing a single-channel 257×188 representation.

Video is represented by 3 RGB frames sampled at 1 fps, resized to 224×224, and normalized using ImageNet statistics. Training-time augmentation uses random resized crops and random horizontal flips; validation and test use deterministic resizing.

H.4 Model Architecture

Each modality is encoded by a ResNet18 backbone adapted for its input shape. The audio backbone takes the single-channel spectrogram and applies a global average pool to produce a 512-dimensional representation; the video backbone takes the 3-frame clip and applies a 3D global average pool to produce a 512-dimensional representation. Both backbones are pretrained unimodally on CREMA-D before joint training, with the same pretrained checkpoints used as initialization across all methods.

The fusion trunk concatenates the two 512-dimensional encoder outputs and applies $\text{Linear}(1024 \to 64) \to \text{ReLU} \to \text{Dropout}(0.1)$. The fusion head is a single $\text{Linear}(64 \to 7)$ producing the 7-way classification logits. Auxiliary unimodal heads ($\text{Linear}(512 \to 64) \to \text{ReLU} \to \text{Dropout}(0.1) \to \text{Linear}(64 \to 7)$) are attached to each encoder for the SynIB inner loop. All baselines share the same ResNet18 backbones and pretrained initialization; they differ in the fusion mechanism according to each method's specification.

H.5 Training and SynIB Configuration

We train with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$), learning rate 10^{-5}, weight decay 10^{-5}, and batch size 64, with cosine annealing and a 780-step warmup. Maximum training length is 1500 epochs with early stopping on validation accuracy (patience 30 epochs). The classification loss is unweighted cross-entropy across all 7 classes, with no per-class reweighting.

For SynIB, we use random masking on each modality with corruption probability π and the default uniform reference distribution (App. C.3). The regularization strength is λ. SynIB-specific values are held constant across all α for a given run.

H.6 Evaluation

Models are evaluated on the test split of each fold using two metrics: macro F1 across all 7 classes (reported as Total F1 in Figure 4) and per-class F1 for the irony class (reported as Irony F1). Both metrics are computed on the full test set, which contains $\lfloor \alpha \bar{N}_{\text{test}} \rfloor$ irony samples generated by the same mutation procedure used at training time. Reported values in Figure 4 are mean and standard deviation across 3 folds.

H.7 Numerical Results

Table 4 reports the per-method F1 scores summarized in Figure 4, with standard deviations across folds. Each cell shows irony-class F1 over total F1 (mean $\pm$ standard deviation across 3 folds); bold marks the best entry per column for each metric. Uni-Video and Uni-Audio do not predict the irony class and report total F1 only. Reported values cover the four irony rates used in the main figure ($\alpha \in \{0.1, 0.5, 1.0, 2.0\}$).

Reading the table. SynIB achieves the highest irony-class F1 in the two regimes where synergistic samples are scarce ($\alpha \in \{0.1, 0.5\}$), with margins of $+2.2$ and $+3.4$ points over the best balancing baseline (MCR) and consistently larger margins over the unimodal-rebalancing methods (D&R, MMPareto, ReconBoost). When synergistic samples become abundant ($\alpha \in \{1.0, 2.0\}$), simpler

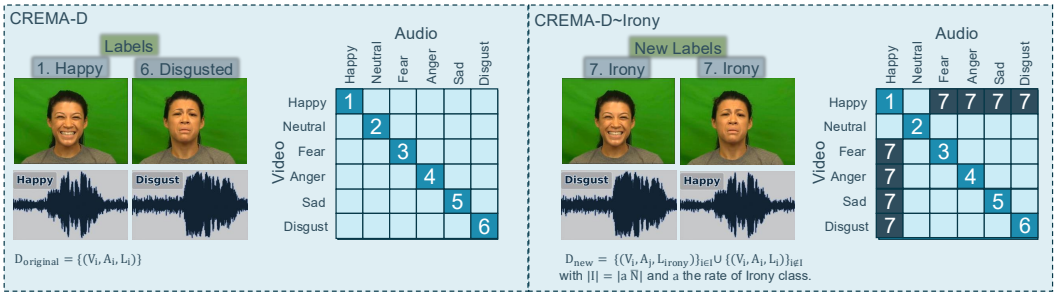


Figure 12: Construction of the synthetic irony class. Each ironic sample is built by pairing a video V_i from one base emotion (e.g., happy) with a donor audio A_j from a contradicting emotion (e.g., disgust). The new label is irony. The dataset is $D_{\text{new}} = \{(V_i, A_j, L_{\text{irony}})\}_{i \in I} \cup \{(V_i, A_i, L_i)\}_{i \notin I}$, with $|I| = \lfloor \alpha \bar{N} \rfloor$ controlled by the irony-rate hyperparameter α. Neither modality alone identifies an ironic sample; the label requires reading the cross-modal contradiction.

Table 4: F1 scores on the CREMA-D irony recognition task under varying irony rates α, reported as irony-class F1 / total F1 (mean $\pm$ standard deviation across 3 folds). Bold marks the best per column for each metric. Dashes indicate that the corresponding metric does not apply.

Irony rate α	0.1	0.5	1.0	2.0
Uni-Video	—/41.0±3.5	—/39.9±2.9	—/38.3±3.7	—/28.0±14.7
Uni-Audio	—/49.8±1.7	—/46.9±2.4	—/45.7±1.3	—/39.6±1.2
Ensemble	3.0±5.2/59.7±4.3	15.0±8.0/58.4±2.8	36.8±6.3/59.6±3.2	54.3±3.8/38.3±21.4
Vanilla fusion	6.4±5.6/59.3±3.2	18.3±12.2/57.4±3.9	39.7±0.2/58.2±1.0	50.0±5.5/54.5±4.6
D&R	0.0±0.0/55.1±1.4	9.0±6.1/55.7±1.6	25.8±9.0/54.3±1.6	45.2±5.9/47.0±0.7
MMPareto	3.7±6.4/58.9±2.3	13.0±4.2/58.6±0.6	20.0±0.8/55.5±1.1	41.5±2.6/50.2±1.8
ReconBoost	0.0±0.0/57.6±3.1	11.6±4.4/57.0±2.0	19.3±7.6/55.2±3.6	45.6±10.0/48.8±2.8
MCR	14.1±12.2/63.7±0.7	17.6±8.7/60.8±2.7	31.6±5.2/59.6±2.8	53.5±6.4/55.5±3.6
SynIB	16.3±6.6/60.4±2.1	21.7±4.7/59.4±2.4	35.3±3.7/57.4±1.7	53.0±3.6/55.3±2.2

methods that do not regularize fusion explicitly (vanilla fusion at $\alpha = 1.0$, ensemble at $\alpha = 2.0$) match or slightly exceed SynIB on irony F1, reflecting that abundant synergy is recoverable without an explicit synergy-targeted objective. SynIB nevertheless remains within 1.3–4.4 points of the leader in these regimes and outperforms all four balancing methods on irony F1 at every α.

On total F1. On macro F1 across all 7 classes, MCR achieves the best score in three of four cells; SynIB stays within 0.2 to 3.3 points of MCR across all rates and outperforms vanilla fusion at $\alpha = 0.1, 0.5$. Ensemble achieves the best total F1 at $\alpha = 1.0$ but collapses at $\alpha = 2.0$ (38.3 ± 21.4), where the high standard deviation reflects fold-level instability rather than a stable improvement. The trade-off between irony-class F1 and total F1 is small and consistent: SynIB sacrifices roughly 1–3 points of total F1 to gain 2–5 points on the synergy-required class.

Variance considerations. Standard deviations on the irony-class metric are large for several baselines (MCR ± 12.2 at $\alpha = 0.1$, vanilla fusion ± 12.2 at $\alpha = 0.5$, ReconBoost ± 10.0 at $\alpha = 2.0$), indicating fold-to-fold instability in their irony predictions when synergistic structure is present but scarce. SynIB's standard deviations on irony F1 are uniformly smaller (between ± 3.6 and ± 6.6), suggesting that the synergy-targeted objective produces more stable irony detection across folds in addition to higher mean F1.

I Per-Method Numerical Results on MultiBench, Hateful Memes, and MM-IMDb

Table 5 reports the per-method numbers underlying Fig. 5 in the main text. Synergy-subset examples are test samples misclassified by every unimodal predictor (see Sec. 4.1 for definition and subset sizes per benchmark). Each cell is the mean over 3 seeds $\pm$ one standard deviation. **Bold** marks the best per column.

Table 5: Synergy-subset accuracy and whole-test accuracy (%) on five real-world multimodal benchmarks. Each cell is the mean over 3 seeds $\pm$ one standard deviation. **Bold** marks the best per column.

Method	UR-Funny	MUStARD	MOSI	Hateful Memes	MM-IMDb
Synergy-subset accuracy (%)					
Ensemble	0.9±1.0	45.1±10.0	0.0±0.0	29.6±0.5	2.3±1.6
Vanilla Fusion	19.2±3.5	25.4±2.5	31.4±4.3	33.8±9.8	21.4±1.2
D&R	15.2±2.3	16.5±11.5	24.1±4.0	42.1±5.7	24.4±3.5
MMPareto	15.9±3.3	40.1±13.4	25.1±5.4	42.4±5.2	24.9±1.6
ReconBoost	12.7±5.9	17.2±6.7	30.2±5.0	42.3±6.1	18.7±0.8
MCR	11.0±1.1	39.6±22.4	29.0±3.1	45.9±6.4	20.9±3.1
CE+InfMasking	10.5±6.2	46.5±12.3	15.7±2.5	42.0±3.0	19.7±1.6
SynIB M_{R}	15.4±3.2	47.5±17.4	35.3±3.9	**48.9**±4.0	24.6±0.6
SynIB M_{L}	**22.8**±6.2	**52.9**±7.7	**37.2**±5.0	47.0±1.8	**25.6**±2.4
Whole-test accuracy (%)					
Ensemble	61.3±3.9	58.0±3.0	72.8±1.6	55.2±0.5	83.5±0.1
Vanilla Fusion	62.3±0.1	57.5±2.3	72.8±1.8	66.4±2.3	83.5±0.2
D&R	**63.6**±0.5	54.5±5.1	73.0±1.5	67.2±1.3	83.4±0.9
MMPareto	62.8±0.1	58.8±2.2	72.6±1.9	68.0±1.7	83.7±0.2
ReconBoost	63.2±0.3	57.5±5.6	74.7±1.7	66.9±1.7	84.6±0.1
MCR	62.4±0.2	60.3±8.1	73.7±1.3	68.7±1.2	83.8±0.1
CE+InfMasking	63.1±0.3	61.9±5.2	71.9±1.3	66.9±1.7	83.7±0.8
SynIB M_{R}	**63.6**±0.6	62.2±1.9	75.0±1.8	**69.8**±1.0	**85.1**±0.4
SynIB M_{L}	62.7±0.3	**64.1**±3.0	**75.3**±1.1	69.0±1.1	85.0±0.3

Reading the table. On the synergy subset, a SynIB variant is the top method on every benchmark: M_{L} on UR-Funny, MUStARD, and MOSI, and M_{R} on Hateful Memes; M_{R} trails the strongest baseline only on UR-Funny (-3.8) and MM-IMDb (-0.3). No baseline is consistently strongest: vanilla fusion leads the baselines on UR-Funny and MOSI, CE+InfMasking on MUStARD, and MCR on Hateful Memes. Relative to vanilla fusion, CE+InfMasking gains on MUStARD ($+21.1$) and Hateful Memes ($+8.2$) but loses on UR-Funny (-8.7), MOSI (-15.7), and MM-IMDb (-1.7); both SynIB variants exceed it on all five synergy subsets, and M_{R} also on whole-test accuracy ($+0.3$ to $+3.1$). On the full test set, a SynIB variant is best on MUStARD, MOSI, and Hateful Memes and ties D&R on UR-Funny, where M_{L} trails by 0.9 points, the only whole-test deficit of either variant. Synergy-subset standard deviations are large for most methods on MUStARD (up to ± 22.4 for MCR and ± 17.4 for SynIB M_{R}), reflecting its small test set, so differences there should be read against this spread.

MM-IMDb. In Table 5, MMPareto is the strongest baseline on the synergy subset (24.9) and ReconBoost on the full test set (84.6). SynIB M_{L} is best on the synergy subset ($+0.7$), while M_{R} is 0.3 points below MMPareto. On the full test set, both variants lead ($+0.5$ and $+0.4$). These margins are small, but the paired tests over 10 runs (App. I.6) find both variants significantly better than every baseline on whole-test accuracy.

I.1 Alternative synergy-subset definitions

The synergy subset of Sec. 4.1 contains the test samples misclassified by every unimodal model. Since unimodal models can also miss a sample because of label noise or weak unimodal predictors, we rebuild the subset under six alternative definitions and re-score all methods on each (Table 6). Three use several independently seeded unimodal models per modality, keeping the samples misclassified by

Table 6: Synergy-subset accuracy (%) under our definition (Orig., identical to Table 5) and six alternative definitions (App. I.1). All columns are computed from the same trained models; the subsets differ in size and difficulty, so compare methods within a column rather than values across columns. Bold marks the best method per column within each benchmark (ties included). The late ensemble and CE+InfMasking were not re-scored.

		Based on unimodal models					Model-free
Method	Orig.	Strict	Major.	Ensem.	Unc.	Conf.	CVX-PID
Hateful Memes							
Vanilla fusion	33.8	36.4	42.3	43.5	50.6	22.0	64.5
MCR	45.9	41.5	48.1	49.1	50.9	**32.1**	66.4
D&R	42.1	36.9	43.0	43.5	45.1	28.5	65.0
MMPareto	42.4	42.4	48.3	48.0	52.6	**32.1**	65.9
ReconBoost	42.3	39.1	43.4	44.0	49.4	28.8	63.9
SynIB M_{R}	**48.9**	**44.7**	49.9	50.4	**57.4**	31.8	**69.8**
SynIB M_{L}	47.0	44.0	**50.1**	**51.1**	56.6	31.3	68.1
MUStARD							
Vanilla fusion	25.4	25.5	37.5	37.6	29.6	20.8	55.6
MCR	39.6	41.2	47.5	47.0	48.1	33.3	57.4
D&R	16.5	19.6	23.3	23.1	25.9	12.5	59.3
MMPareto	40.1	37.3	37.5	36.8	40.7	33.3	53.7
ReconBoost	17.2	7.8	15.8	14.5	3.7	12.5	**61.1**
SynIB M_{R}	47.5	39.2	44.2	44.4	48.1	29.2	57.4
SynIB M_{L}	**52.9**	**45.1**	**48.3**	**48.7**	**51.9**	**37.5**	**61.1**
UR-Funny							
Vanilla fusion	19.2	**15.7**	23.0	23.3	15.6	**15.8**	**65.1**
MCR	11.0	4.6	10.2	10.4	8.1	1.1	61.9
D&R	15.2	12.7	20.9	20.7	16.7	8.7	62.2
MMPareto	15.9	7.9	17.6	17.7	9.1	6.6	63.0
ReconBoost	12.7	7.9	14.7	14.1	10.8	4.9	61.1
SynIB M_{R}	15.4	4.3	12.3	13.4	6.5	2.2	61.9
SynIB M_{L}	**22.8**	15.2	**23.3**	**24.0**	**20.4**	9.8	61.4

every seeded model (Strict), by a majority of them (Major.), or by their softmax ensemble (Ensem.). Two split unimodal errors by confidence into low-confidence (Unc.) and high-confidence (Conf.) errors. The sixth, CVX-PID, uses no unimodal model at all: it applies the convex PID estimator of Liang et al. [2023] and keeps the test samples in the highest-synergy cells. As a check against ground truth, on CREMA-D-Irony, where synergy is known by construction, 90.7% of the ironic test samples fall in our synergy subset.

On MUStARD, SynIB M_{L} is the best method under every definition, tied with ReconBoost under CVX-PID. On Hateful Memes, a SynIB variant is best under five of the six; under Conf., MCR and MMPareto lead by 0.3 points (32.1 vs. 31.8). UR-Funny is the exception: SynIB M_{L} leads under Major., Ensem., and Unc., but vanilla fusion leads under Strict (15.7 vs. 15.2), Conf. (15.8 vs. 9.8), and CVX-PID (65.1 vs. 61.9). Overall, a SynIB variant is best or tied in 14 of the 18 alternative columns. On Hateful Memes and MUStARD, the advantage seen in Fig. 5 thus survives every alternative definition, including the model-free one, apart from the 0.3-point gap under Conf.; on UR-Funny, it depends on the definition. MOSI is omitted, as its subset of 17 samples is too small to subdivide.

I.2 Sensitivity to the SynIB weight λ

We probe how SynIB depends on its regularisation weight λ, which scales the synergy term relative to the task loss. For every benchmark we sweep $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$ with the learned mask and otherwise fix each dataset's configuration. Each point is the mean over n=3 runs (the three cross-validation folds for the MultiBench datasets; three seeds for Hateful Memes), and we report both *whole-test accuracy* and *synergy-subset accuracy*, with ± 1 standard deviation. Fig. 13 shows the result; the dashed line marks the value selected on the validation set.

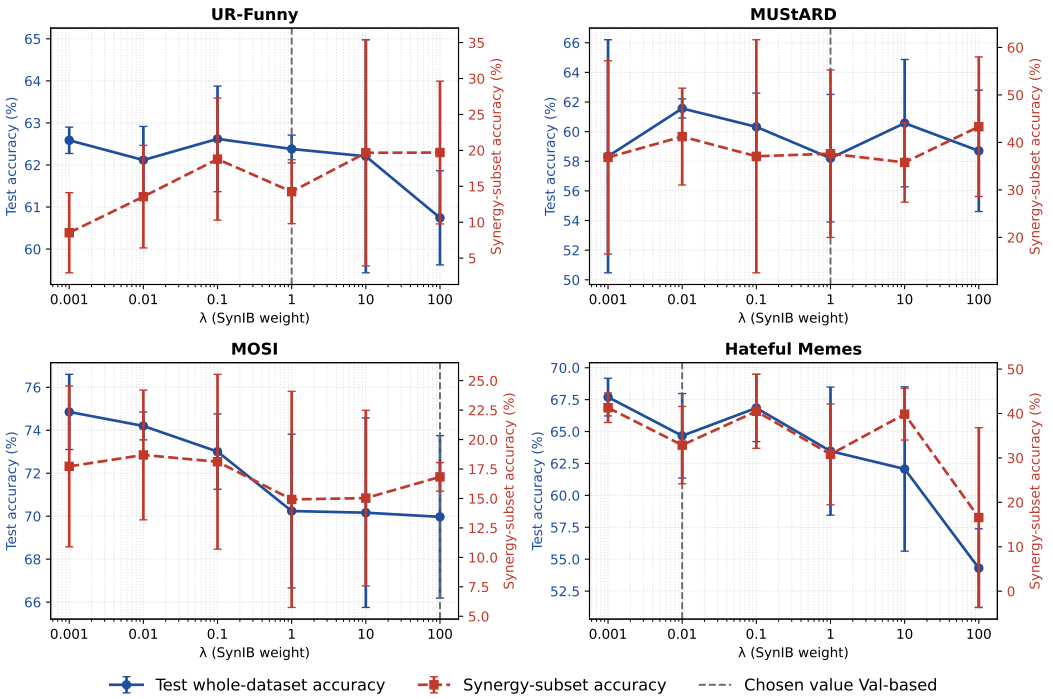


Figure 13: Sensitivity of SynIB to the weight λ across the four benchmarks. Blue: whole-test accuracy (left axis); red: synergy-subset accuracy (right axis); shaded bars are ± 1 standard deviation over $n{=}3$ runs. The dashed vertical line is the λ chosen on validation. Accuracy is broadly stable across log values of λ on the MultiBench datasets; on Hateful Memes large λ is harmful. We select λ per dataset to maximise validation synergy.

On UR-Funny, MUStARD and MOSI, whole-test accuracy varies by at most 6 points across the entire sweep (Fig. 13), and MOSI loses about 5 of them once $\lambda \geq 1$: there is no sharp optimum, only an upper limit on λ, so SynIB does not require careful tuning of λ to remain competitive. The flat response also indicates that the synergy term acts as a mild, well-behaved regulariser in this regime rather than a destabilising force.

Only very large λ degrades performance, and only where the loss is on a different scale. The exception is Hateful Memes, where accuracy stays within about 2 points of its best up to $\lambda = 0.1$, declines from $\lambda = 1$ (63.3 vs. 69.0), and falls steeply at $\lambda = 10^2$ (54.3) as the synergy term overwhelms the task loss. This boundary, together with the fact that the validation-selected λ differs across datasets (e.g. small for Hateful Memes, larger for MOSI), simply reflects that the synergy term is not normalised to the per-dataset loss magnitude; the *shape* of the curve, plateau followed by a high-λ drop, is consistent across benchmarks.

I.3 Learned-mask hyperparameters

Learned masking adds two hyperparameters: the sparsity weight λ_M of Eq. (5) and the number of inner-loop steps that update the mask at each training step. We vary each one at a time around each dataset's default, with all other settings fixed, and report whole-test and synergy-subset accuracy together with the masked fraction, the share of feature dimensions the learned mask corrupts, averaged over training (Table 7). The masked fraction shows whether a setting changes the mask, separately from whether it changes the result.

On MUStARD and MOSI, raising λ_M from 0.01 to 10 shrinks the masked fraction from 40% to 6%, yet whole-test accuracy varies by at most 6.1 points on MUStARD and 4.8 on MOSI, and by at most 1.7 on UR-Funny and 2.1 on Hateful Memes. Synergy-subset accuracy varies more on MOSI and MUStARD, whose subsets are small (17 test examples on MOSI), so a swing there corresponds to a few examples. For the inner loop, 5 steps lowers synergy-subset accuracy on MUStARD and

MOSI relative to the default of 20, while 50 steps changes the mask (2% masked on MUStARD, 47% on MOSI) with whole-test accuracy within 2.2 points. The only degenerate configuration we observed combines 50 inner steps with a large λ_M on MUStARD and MOSI: the masked fraction falls to 1–2%, so SynIB reduces to training without masking, and accuracy is unaffected. Learning rate and batch size, varied in the same way, change whole-test accuracy by at most 3.1 and 1.2 points.

Table 7: Sensitivity to the learned-mask hyperparameters: the sparsity weight λ_M (left) and the number of inner-loop steps that update the mask at each training step (right), each varied around the dataset's default (bold) with all else fixed. Whole-test (Whole) and synergy-subset (Syn.) accuracy, mean ± sd over 3 runs, and the percentage of feature dimensions the learned mask corrupts, averaged over training (Masked). Inner-loop steps were not swept on Hateful Memes.

		Sparsity weight λ_M					Inner-loop steps		
Benchmark		0.01	0.1	1	10	100	5	20	50
MUStARD	Whole	60.5±2.2	62.3±2.8	58.0±5.6	**64.1**±3.0	60.8±3.3	60.1±4.9	**64.1**±3.0	62.6±0.9
	Syn.	40.6±14.8	50.2±6.9	40.1±17.4	**52.9**±7.7	47.9±8.9	45.8±13.5	**52.9**±7.7	55.4±14.9
	Masked (%)	40	25	11	**6**	6	20	**6**	2
MOSI	Whole	73.3±2.6	**75.3**±1.1	73.9±2.3	70.5±1.5	72.0±2.0	72.0±2.5	**75.3**±1.1	73.1±1.8
	Syn.	16.2±6.4	**37.2**±5.0	15.4±5.1	13.1±4.6	15.6±2.0	10.4±4.4	**37.2**±5.0	19.3±6.5
	Masked (%)	40	**31**	10	6	6	26	**31**	47
UR-Funny	Whole	**62.7**±0.3	62.9±1.2	62.6±0.9	61.8±0.4	61.2±1.1	62.6±1.6	**62.7**±0.3	62.2±0.4
	Syn.	**22.8**±6.2	15.6±11.3	11.3±8.1	17.6±14.8	17.5±9.6	21.0±5.7	**22.8**±6.2	19.6±11.4
	Masked (%)	**21**	12	8	6	6	25	**21**	31
Hateful Memes	Whole	66.9±3.2	**69.0**±1.1	67.3±1.0	67.7±1.7	67.5±1.1			
	Syn.	42.3±8.8	**47.0**±1.8	45.4±7.1	44.5±3.0	43.3±3.7		–	
	Masked (%)	26	**23**	21	20	20			

I.4 Masking rate π of random masking

Random masking corrupts each coordinate independently with probability π (Sec. 3.5), tuned per dataset: $\pi = 0.7$ on UR-Funny, 0.1 on MUStARD, 0.3 on MOSI, Hateful Memes, MM-IMDb, and CREMA-D-Irony, 0.4 on Spurious XOR, and 0.5 on PID-XOR. Learned masking has no π.

To check sensitivity, we swept $\pi \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ on CREMA-D-Irony and MUStARD with all other settings fixed. Across the grid, mean accuracy varies by 1.4 points on CREMA-D-Irony, with no setting differing from $\pi = 0.5$ by more than 1.5 points paired within fold, and by 3.5 points on MUStARD, within fold-to-fold variation. No value causes a sharp drop: although π is tuned per dataset, the results do not hinge on it, and random masking degrades gracefully across the range.

I.5 Larger and pretrained multimodal backbones

Table 1a re-runs every method in Table 5 on Hateful Memes under three additional configurations. Two scale the encoders, pairing CLIP-L/14 with DeBERTa-v3-large (740M parameters) or with Qwen2.5-1.5B (1.8B). The third replaces our fusion Transformer with FLAVA's pretrained multimodal network, using its ViT-B/16 image and BERT-base text towers and its pretrained 6-layer fusion encoder (43M of the model's 241M parameters). The new configurations report accuracy at the best validation step; the late ensemble and CE+InfMasking were not run at them. Two further observations: the margin of SynIB M_{R} over vanilla fusion on the synergy subset narrows as capacity grows, from 15.1 points in the base configuration to 5.7 at 740M and 3.4 at 1.8B, while staying positive throughout; and on whole-test accuracy, SynIB M_{R} is best at 740M and within 0.6 and 1.2 points of the best method with FLAVA and at 1.8B.

I.6 Statistical significance

We report the mean paired difference Δ between a SynIB variant and a baseline, in accuracy points, with a 95% confidence interval from the Nadeau–Bengio corrected t-test [Nadeau and Bengio, 2003]; p-values are Holm-adjusted [Holm, 1979] across the baselines within each benchmark, metric, and variant. On MM-IMDb, both variants improve whole-test accuracy over all five baselines, and SynIB

Table 8: Paired comparison of whole-test accuracy over the 10 runs of App. I.6. Δ: mean paired difference in points (SynIB minus baseline), with a 95% confidence interval (Nadeau–Bengio corrected, not adjusted for multiple comparisons). p_{Holm}: Holm-adjusted across the baselines within each benchmark and variant. Bold: $p_{\text{Holm}} < 0.05$. Hateful Memes was not compared with MCR.

		SynIB M_{R}		SynIB M_{L}	
Benchmark	vs.	Δ [95% CI]	p_{Holm}	Δ [95% CI]	p_{Holm}
MM-IMDb	Vanilla	$+1.98$ [$+1.11$, $+2.86$]	**0.003**	$+1.88$ [$+1.27$, $+2.48$]	**<0.001**
	D&R	$+1.70$ [$+0.75$, $+2.65$]	**0.006**	$+1.59$ [$+0.75$, $+2.43$]	**0.004**
	MMPareto	$+1.91$ [$+1.07$, $+2.75$]	**0.003**	$+1.80$ [$+1.15$, $+2.45$]	**<0.001**
	ReconBoost	$+0.93$ [$+0.23$, $+1.62$]	**0.014**	$+0.82$ [$+0.18$, $+1.46$]	**0.018**
	MCR	$+1.41$ [$+0.81$, $+2.02$]	**0.003**	$+1.31$ [$+0.72$, $+1.89$]	**0.002**
MUStARD	Vanilla	$+2.02$ [-1.53, $+5.56$]	0.280	$+1.05$ [-5.08, $+7.18$]	0.874
	D&R	$+7.15$ [$+2.17$, $+12.13$]	0.050	$+6.19$ [$+1.82$, $+10.56$]	0.055
	MMPareto	$+7.22$ [$+0.39$, $+14.04$]	0.160	$+6.25$ [$+0.32$, $+12.18$]	0.164
	ReconBoost	$+8.65$ [-1.10, $+18.41$]	0.228	$+7.69$ [-1.01, $+16.39$]	0.231
	MCR	$+3.51$ [-1.40, $+8.42$]	0.280	$+2.54$ [-4.54, $+9.63$]	0.874
Hateful Memes	Vanilla	$+0.68$ [-0.06, $+1.43$]	0.272	$+0.02$ [-1.78, $+1.81$]	1.000
	D&R	$+0.95$ [-0.45, $+2.34$]	0.483	$+0.20$ [-1.62, $+2.01$]	1.000
	MMPareto	-0.19 [-1.51, $+1.14$]	1.000	-0.94 [-2.32, $+0.45$]	0.640
	ReconBoost	$+0.84$ [-2.10, $+3.78$]	1.000	$+0.09$ [-2.81, $+2.99$]	1.000
UR-Funny	Vanilla	-0.76 [-1.68, $+0.15$]	0.460	-0.32 [-1.24, $+0.60$]	1.000
	D&R	-0.15 [-0.42, $+0.12$]	0.916	$+0.29$ [-0.52, $+1.11$]	1.000
	MMPareto	-1.18 [-3.24, $+0.89$]	0.916	-0.74 [-2.84, $+1.37$]	1.000
	ReconBoost	$+0.87$ [-1.34, $+3.08$]	0.916	$+1.31$ [-1.12, $+3.74$]	1.000
	MCR	$+0.45$ [-0.94, $+1.84$]	0.916	$+0.90$ [-0.70, $+2.49$]	1.000
MOSI	Vanilla	-0.91 [-4.08, $+2.26$]	1.000	-1.16 [-4.33, $+2.01$]	1.000
	D&R	$+0.38$ [-3.22, $+3.98$]	1.000	$+0.13$ [-4.75, $+5.02$]	1.000
	MMPareto	$+0.12$ [-3.81, $+4.05$]	1.000	-0.13 [-3.94, $+3.68$]	1.000
	ReconBoost	$+1.09$ [-3.14, $+5.33$]	1.000	$+0.84$ [-3.64, $+5.33$]	1.000
	MCR	-0.73 [-4.60, $+3.15$]	1.000	-0.97 [-4.00, $+2.05$]	1.000

M_{R} improves the synergy subset over three of them. On the four smaller benchmarks, no difference is significant after correction except on the MOSI synergy subset, where D&R exceeds SynIB M_{R}; Sec. 4.1 already flags this subset as too small for firm conclusions. We test significance with paired comparisons over 10 random train/test splits of each benchmark (Tables 8 and 9).

I.7 Compute Resources

All experiments ran on NVIDIA A100 GPUs (40 GB HBM2). Each job was allocated a single A100, 8 CPU cores, and 80 GB of host RAM. Final-result runs (3 seeds $\times$ 8 methods $\times$ 4 datasets, 96 runs total) consumed roughly 200 GPU-h. Hyperparameter sweeps for SynIB and the four balancing baselines added approximately 1000 GPU-h, the bulk on Hateful Memes due to its larger grid and longer per-run time. The full main-paper table can be reproduced end-to-end on a single A100 in roughly 50 wall-clock hours. All runs use Python 3.10, PyTorch 2.1, CUDA 12.1, and HuggingFace `transformers` 4.38; checkpoints, logs, and reproduction scripts are released with the code.

J Baselines

We compare SynIB against representative multimodal learning strategies spanning unimodal models, standard fusion, and methods designed to mitigate multimodal competition. All baselines use the same backbone architecture and differ only in their optimization objectives or training procedures.

Let $(x_1, x_2, y) \sim p(x_1, x_2, y)$ denote multimodal samples and let $f_\theta(x_1, x_2) \in \Delta(\mathcal{Y})$ be the joint predictor. Unimodal predictors are denoted $f_{\theta_1}(x_1)$ and $f_{\theta_2}(x_2)$.

Table 9: Paired comparison of synergy-subset accuracy over the same 10 runs; columns as in Table 8.

		SynIB M_{R}		SynIB M_{L}	
Benchmark	vs.	Δ [95% CI]	p_{Holm}	Δ [95% CI]	p_{Holm}
MM-IMDb	Vanilla	$+4.42$ [$+1.28$, $+7.56$]	**0.033**	$+3.70$ [$+0.19$, $+7.20$]	0.123
	D&R	$+3.09$ [-0.81, $+6.99$]	0.214	$+2.37$ [-2.62, $+7.35$]	0.622
	MMPareto	$+2.04$ [-1.63, $+5.70$]	0.240	$+1.32$ [-2.73, $+5.36$]	0.622
	ReconBoost	$+5.65$ [$+2.34$, $+8.95$]	**0.020**	$+4.92$ [$+1.07$, $+8.78$]	0.072
	MCR	$+5.50$ [$+2.12$, $+8.87$]	**0.020**	$+4.78$ [$+1.28$, $+8.28$]	0.065
MUStARD	Vanilla	$+9.15$ [-11.02, $+29.32$]	0.996	$+1.30$ [-19.64, $+22.24$]	1.000
	D&R	$+2.12$ [-11.87, $+16.11$]	0.996	-5.72 [-21.35, $+9.91$]	1.000
	MMPareto	$+9.69$ [-6.99, $+26.36$]	0.884	$+1.84$ [-13.55, $+17.23$]	1.000
	ReconBoost	$+7.40$ [-15.55, $+30.36$]	0.996	-0.44 [-24.37, $+23.48$]	1.000
	MCR	$+12.71$ [-1.93, $+27.35$]	0.405	$+4.86$ [-8.84, $+18.57$]	1.000
Hateful Memes	Vanilla	$+2.60$ [-0.56, $+5.75$]	0.376	$+0.88$ [-5.75, $+7.50$]	1.000
	D&R	$+2.45$ [-1.42, $+6.31$]	0.558	$+0.38$ [-6.06, $+6.82$]	1.000
	MMPareto	$+0.98$ [-2.75, $+4.70$]	1.000	-1.09 [-6.54, $+4.37$]	1.000
	ReconBoost	$+2.55$ [-7.26, $+12.37$]	1.000	$+0.49$ [-9.50, $+10.47$]	1.000
UR-Funny	Vanilla	$+2.62$ [-1.10, $+6.33$]	0.700	-0.61 [-6.74, $+5.53$]	1.000
	D&R	-1.20 [-8.68, $+6.28$]	1.000	-2.54 [-9.38, $+4.30$]	1.000
	MMPareto	-3.70 [-14.44, $+7.04$]	1.000	-5.04 [-14.23, $+4.15$]	0.984
	ReconBoost	-5.19 [-13.88, $+3.50$]	0.840	-6.53 [-14.03, $+0.97$]	0.400
	MCR	$+2.74$ [-4.56, $+10.04$]	1.000	-0.48 [-7.03, $+6.06$]	1.000
MOSI	Vanilla	$+2.33$ [-3.39, $+8.06$]	0.762	$+5.37$ [-2.44, $+13.18$]	0.592
	D&R	-18.84 [-31.43, -6.24]	**0.040**	-15.80 [-33.27, $+1.67$]	0.355
	MMPareto	-12.08 [-24.80, $+0.64$]	0.180	-9.04 [-21.98, $+3.90$]	0.592
	ReconBoost	-10.24 [-19.17, -1.30]	0.116	-7.20 [-20.47, $+6.07$]	0.592
	MCR	-2.64 [-18.71, $+13.43$]	0.762	$+0.40$ [-12.01, $+12.80$]	0.944

Unimodal models. Audio-only and video-only classifiers are trained independently using cross-entropy: $\mathcal{L}_{\text{uni},i} = \mathcal{L}_{\text{task}}(f_{\theta_i}(y \mid x_i), Y) \quad i \in \{1, 2\}$. These models establish a lower bound and verify whether the task is solvable from a single modality.

Late ensemble. Unimodal predictions are averaged at inference: $f_{\text{ens}}(y \mid x_1, x_2) = \frac{1}{2}\left(f_{\theta_1}(y \mid x_1) + f_{\theta_2}(y \mid x_2)\right)$. No joint representation learning occurs. This baseline tests whether aggregation alone can recover cross-modal signal.

Vanilla fusion. Standard multimodal fusion minimizes cross-entropy on joint inputs $\mathcal{L}_{\text{fusion}} = \mathcal{L}_{\text{task}}(f_\theta(y \mid x_1, x_2), Y)$. This objective does not constrain modality contributions and is known to exhibit multimodal competition, where optimization favors dominant unimodal shortcuts Huang et al. [2022].

Diagnosing & Re-learning (D&R) Wei et al. [2024]. D&R mitigates modality collapse by periodically reinitializing modality encoders when imbalance is detected. Let g_k denote the purity-gap diagnostic for modality k. The reset strength is $\alpha_k = \tanh(\lambda g_k)$. When imbalance exceeds a threshold, parameters are softly reset via $\theta_k \leftarrow (1 - \alpha_k)\theta_k + \alpha_k \theta_k^{\text{init}}$. Training then resumes under the vanilla fusion objective.

D&R prevents one modality from monopolizing learning by periodically restoring weaker modality capacity. This stabilizes the balance of unique contributions across modalities, but it does not introduce an objective that favors joint-only information. Consequently, D&R addresses optimization imbalance without explicitly promoting synergistic representations.

MMPareto Wei and Hu [2024]. MMPareto replaces the vanilla fusion gradient with a Pareto-integrated gradient that balances unimodal and multimodal objectives. Let $\mathcal{L}_{12} = \mathcal{L}(x_1, x_2, y), \quad \mathcal{L}_1 = \mathcal{L}(x_1, y), \quad \mathcal{L}_2 = \mathcal{L}(x_2, y)$ denote multimodal and unimodal losses. The parameter update uses the integrated gradient $g^* = \sum_{m \in \{1,2,12\}} w_m \nabla \mathcal{L}_m$, where the weights are

obtained by solving the Pareto min-norm problem

$$\min_{\mathbf{w}\in\Delta} \left\| \sum_m w_m \nabla \mathcal{L}_m \right\|_2, \quad \Delta = \{w_m \geq 0, \sum_m w_m = 1\}. \tag{23}$$

Parameters are updated using g^* instead of the vanilla gradient.

From a PID perspective, MMPareto equalizes optimization pressure across unimodal and multimodal objectives, preventing one modality from dominating learning. This stabilizes unique and redundant information usage, but does not explicitly prioritize signals that arise only from joint cross-modal interaction. As a result, it balances modality competition without directly targeting synergy.

ReconBoost Hua et al. [2024]. ReconBoost alternates unimodal updates while enforcing agreement and gradient consistency between modality predictors. The fused prediction is $p_{12} = \sum_{k=1}^{M} p_k$, where $p_k = p(y|x_k)$ are unimodal predictors.At boosting round s, modality k optimizes

$$\tilde{\mathcal{L}}^{\,s} = \mathcal{L}_k - \lambda\, D_{\mathrm{KL}}(p_{-k} \,\|\, p_k) + \alpha \|\nabla \mathcal{L}_k - \nabla \mathcal{L}_{k-1}\|^2, \tag{24}$$

where $\mathcal{L}_k = -\log p_k(y)$ and $p_{-k} = \sum_{j\neq k} p_j$. After alternating rounds, a global rectification step minimizes $\mathcal{L}_{\text{rect}} = -\log p_{12}(y)$.

ReconBoost alternates unimodal specialization while enforcing agreement and gradient alignment between modalities. This stabilizes redundancy and prevents dominance by a single modality, but the objective reconciles unimodal predictors rather than rewarding information that exists only in their interaction. Consequently, ReconBoost mitigates modality conflict without explicitly optimizing for synergistic representations.

MCR (Multimodal Competition Regularizer) Kontras et al. [2025]. MCR augments the fusion loss with a regularizer derived from a mutual-information decomposition:

$$\nabla \mathcal{L} = \nabla \mathcal{L}_{\text{fusion}} + \mathcal{L}_{\text{Con}} + \lambda \Big(\nabla_{\theta_1} \mathcal{L}_{\text{MIPD}_1} - \nabla_{\theta_2} \mathcal{L}_{\text{MIPD}_1} - \nabla_{\theta_1} \mathcal{L}_{\text{MIPD}_2} + \nabla_{\theta_2} \mathcal{L}_{\text{MIPD}_2} \Big). \tag{25}$$

The Mutual Information Perturbed Difference term measures modality contribution via latent perturbations:

$$\mathcal{L}_{\text{MIPD}} = -\mathbb{E}\Big[\mathrm{JSD}(p_{12}, p_{\tilde{1}2}) + \mathrm{JSD}(p_{12}, p_{1\tilde{2}}) \Big], \tag{26}$$

where $p_{12} = p(y|x_1, x_2)$, $p_{\tilde{1}2} = p(y|\tilde{x}_1, x_2)$, and $p_{1\tilde{2}} = p(y|x_1, \tilde{x}_2)$ and the shared task-relevant information is preserved through supervised contrastive alignment $\mathcal{L}_{\text{Con}} = -\log \frac{\psi(z_1, z_2^+)}{\sum_k \psi(z_1, z_{2,k}^-)}$.

MCR regulates how much each modality contributes individually, preventing one modality from dominating the prediction. The perturbation term enforces sensitivity to both inputs, encouraging the model to retain unique modality information, while the contrastive alignment term preserves shared task-relevant structure. However, these mechanisms operate by balancing redundancy and unique contributions rather than explicitly modeling information that exists only in the joint interaction. As a result, MCR mitigates modality collapse but does not directly optimize for synergistic representations.

CE+InfMasking Wen et al. [2025]. InfMasking is a self-supervised contrastive objective: during fusion, it stochastically masks most features of each modality and maximizes the mutual information between the fused representation of the intact pair and those of its masked versions, approximated by the InfMasking loss $\mathcal{L}_{\text{InfM}}$. Since it uses no labels, we port it to our supervised pipeline by adding it to the task loss,

$$\mathcal{L} = \mathcal{L}_{\text{task}}\big(f_\theta(y \mid x_1, x_2), Y\big) + \beta\, \mathcal{L}_{\text{InfM}}, \tag{27}$$

under the same backbones, splits, and training protocol as the other methods. We tune β as for the other baselines and keep all other InfMasking hyperparameters at their published values. InfMasking was designed for pretraining, so this is the closest supervised setting rather than its intended use; we report it on MultiBench, Hateful Memes, and MM-IMDb (Table 5).

From a PID perspective, write $I(X_1, X_2; Y) = R + U_1 + U_2 + S$. Under the standard assumption that contrastive views preserve task-relevant information, the global term of InfMasking targets this aggregate, in which S and R enter with equal weight; its unimodal alignment terms target

$I(X_i; Y) = R + U_i$, in which S does not appear; and its masking term does not involve Y. No term therefore isolates S. For example, XOR ($S = 1$ bit, $R = U_i = 0$) and a copy task in which both modalities carry the label ($R = 1$ bit, $S = 0$) share the same $I(X_1, X_2; Y)$ and hence the same global target. SynIB instead derives its penalty from $I(X_1; Y \mid X_2) = U_1 + S$, where conditioning removes R and separates the two tasks (1 bit versus 0). The two methods also use the mask in opposite directions: InfMasking rewards fused representations whose agreement with the intact view survives masking, whereas SynIB penalizes predictions that survive the removal of a modality's content, which by definition excludes synergy. Since they act on different quantities and at different training stages, the two are complementary: InfMasking can supply pretrained representations on which SynIB is applied downstream.

K NTK Diagnostics for PID Learning

This appendix expands on the NTK-based diagnostic introduced in Sec. 3.1. We restate the core definitions for self-containedness, then provide additional intuition on the quantities involved and discuss the limitations of this analysis.

Setup. Consider a model $f(x_i; \theta)$ with parameters $\theta \in \mathbb{R}^P$, trained on examples (x_i, y_i) grouped by PID source $g \in \{\text{U, Red, Syn}\}$, with $\mathcal{G}$ denoting the index set of group g. As in the main text, we form residuals $s_i^{(g)} = y_i - f(x_i; \theta)$ for $i \in \mathcal{G}$ and define the gradient update direction in parameter space as $v_g = J(\theta)^\top s^{(g)}$, where $J(\theta)_{i,k} = \partial f(x_i; \theta)/\partial \theta_k$ is the Jacobian. The empirical NTK is $K(\theta) = J(\theta)J(\theta)^\top$, with entries $K_{ij} = \nabla_\theta f(x_i; \theta) \cdot \nabla_\theta f(x_j; \theta)$ measuring the inner product between parameter-space gradients of two examples, and thus capturing how coupled their predictions are through the shared parameters.

NTK strength and cosine similarity. The two quantities reported in Sec. 3.1 admit natural interpretations. The NTK strength for group g can be rewritten as the Rayleigh quotient of $K(\theta)$ with respect to $s^{(g)}$:

$$\lambda_g = \frac{\|v_g\|^2}{\|s^{(g)}\|^2} = \frac{s^{(g)\top} K(\theta)\, s^{(g)}}{s^{(g)\top} s^{(g)}},$$

which measures how efficiently the network converts prediction errors from group g into parameter updates, independently of error magnitude. The cosine similarity between update directions,

$$\cos(g, h) = \frac{\langle v_g, v_h \rangle}{\|v_g\|\, \|v_h\|} = \frac{s^{(g)\top} K(\theta)\, s^{(h)}}{\sqrt{s^{(g)\top} K(\theta)\, s^{(g)}} \cdot \sqrt{s^{(h)\top} K(\theta)\, s^{(h)}}},$$

quantifies interference between two PID sources, with near-zero values indicating that they occupy approximately orthogonal subspaces in parameter space with minimal gradient interference.

Limitations. Three caveats are worth keeping in mind when interpreting these results. First, the NTK characterizes training dynamics in the linearized regime where $K(\theta)$ varies slowly [Chizat et al., 2019, Ghorbani et al., 2019], and is therefore best read as a local diagnostic of how the current parameterization processes each PID source rather than a global account of feature learning over training. Second, projecting the NTK along residual-weighted directions $s^{(g)}$ yields group-level summaries that conflate kernel geometry with the current error distribution, meaning that observed differences in λ_g or $\cos(g, h)$ across PID sources may partly reflect differences in error magnitude or sampling frequency rather than intrinsic properties of the kernel. Third, the PID-controlled XOR places each source on disjoint feature blocks and activates one source per example, which biases first-layer parameter gradients toward orthogonal supports across sources; near-zero $\cos(g, h)$ therefore reflects an absence of destructive competition rather than a strong claim of zero coupling, since the design itself favors orthogonality. The strength asymmetry $\lambda_{\text{Syn}} \geq \lambda_{U_1}, \lambda_R$ and the overfitting signature in Fig. 1 are observations about training dynamics that do not depend on this orthogonal block structure.

Our NTK-based analysis should therefore be read as a local diagnostic of gradient interference and learning signal strength, not as a complete characterization of how PID structure is encoded in learned representations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [N/A]

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes],

   Justification: [N/A]

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: [N/A]

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include all the datails needed for reproduction

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We do provide code and our datasets are publicly available.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We do include all the details.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results report the mean and standard deviation over 3 runs; App. I.6 adds paired tests over 10 runs with Nadeau–Bengio corrected confidence intervals and Holm-adjusted p-values.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We do provide details in the appendix.

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed and respected the code of ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We address that in the conclusion and in the limitations.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not provide any large pretraining or any other such large model.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All contributors are authors.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We describe very part of the project in great detail in the appendix.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [No],

    Justification: We did not LLM for the construction of the methodology.