BEYOND A SINGLE REPRESENTATION: STEERING SHARED AND MODALITY-SPECIFIC INFORMATION IN MULTIMODAL SSL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal self-supervised learning aims to learn general representations by exploiting interactions between multiple sources of information. However, most existing approaches learn a single representation with a fixed balance between information shared across modalities and information specific to each modality. We argue that such a fixed operating point might be a limitation for multimodal foundation models, where future downstream tasks are unknown during pretraining and may require different information profiles. Hence, we introduce **STEER**, a preference-conditioned multimodal learning framework that parameterizes a family of representations within a single model. STEER controls the balance between shared and modality-specific information through a preference vector defined on a simplex, and uses low-rank preference-conditioned adaptations to continuously steer the representations toward different information regimes. This allows the desired operating point to be automatically selected after pretraining according to the downstream task, without retraining the representation model. We evaluate STEER in both controlled and real-world multimodal settings. In the former, we show that the learned preference space recovers the expected shared and unique information structure and expands the attainable Pareto region beyond fixed-representation baselines. In the latter, STEER outperforms fixed-representation baselines on several tasks while selecting distinct operating points from the same pretrained model. Importantly, the selected preferences also provide an interpretable description of the balance between modalities favored by each task, which is particularly valuable in scientific applications such as neuroimaging. Code is available here.

1 INTRODUCTION

Learning representations from multiple modalities has become a central problem in self-supervised learning, driven by the idea that different views of the same sample can provide complementary supervision. Contrastive approaches such as CLIP align modality-specific representations in a common space (Radford et al., 2021), while GMC extends this idea by aligning each unimodal representation with a joint multimodal representation (Poklukar et al., 2022). These methods have shown the effectiveness of cross-modal alignment, but they mainly exploit information that is shared across modalities.

This assumption can be restrictive, because useful information is not always shared by all modalities. FactorCL (Liang et al., 2023b) addresses this by factorizing the representation into separate components for shared and modality-specific information. This separation is an important step, but all components are learned jointly under a single fixed pretraining objective. Since these components compete for the same encoder capacity, this objective settles one trade-off between shared and modality-specific signals. A downstream model may concatenate or reweight the components differently, yet it cannot change what information was encoded in them in the first place, nor how representation capacity was allocated between them. Learning a different information balance at the representation level would therefore require changing the training objective and training another model.

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

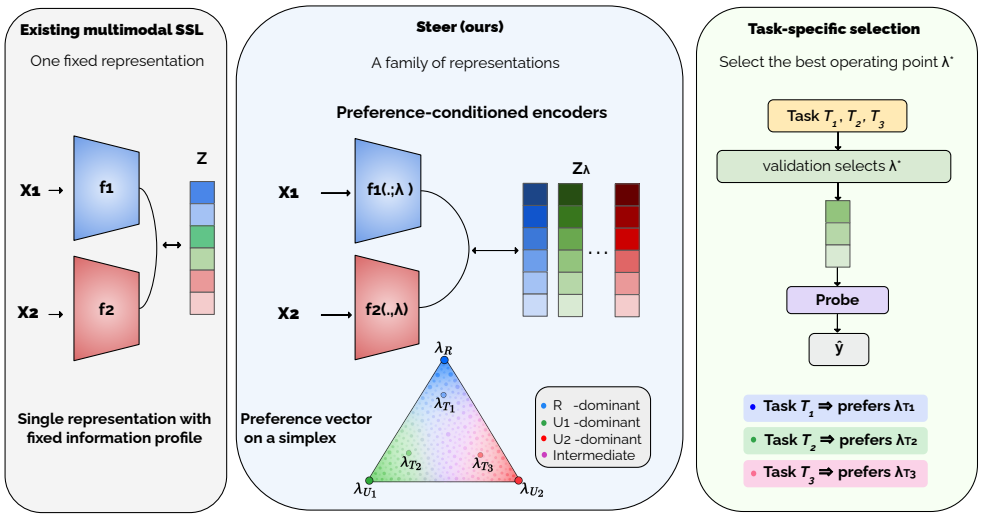


Figure 1: Overview of STEER. Standard multimodal SSL learns a fixed representation, whereas STEER learns a preference-conditioned family spanning shared (R) and modality-specific (U_1, U_2) information. After pretraining, each downstream task selects its preferred operating point without retraining the representation model.

CoMM (Dufumier et al., 2025) takes a different approach. Instead of separating the different information components, it learns a single representation that keeps shared, unique, and synergistic information together. The limitation is that this mixture is fixed during pretraining: once the model is trained, every downstream task receives the same representation, with no direct way to favor more shared or more modality-specific information.

Why should this balance be left to the downstream task? Partial Information Decomposition (PID) (Liang et al., 2023a) gives a useful way to think about this. It decomposes the information that two modalities X_1 and X_2 carry about a target Y into redundant(shared) information R, information unique to each modality U_1 and U_2, and synergistic information S. Importantly, this decomposition depends on the target Y, so it can change from one task to another. A task whose signal is present in both modalities may benefit mostly from shared information, while another task may depend more strongly on one modality. The best balance is therefore task-dependent, but the downstream task is not known during self-supervised pretraining. In this work, we focus on the shared and modality-specific components (R, U_1, U_2) and ask: *can a single pretrained model learn several balances of these information components, and let each downstream task choose the one it needs?*

This question is especially relevant for foundation models, where one pretrained model is expected to support many downstream tasks with different information needs. Instead of fixing one information profile during pretraining, we aim to learn with a single model a family of representations from which each task can select its own operating point, without retraining the representation model (Figure 1).The selected operating point is also useful for interpretation, since it gives an indication of whether a task relies more on shared information or on one modality in particular. This is especially relevant in applications such as neuroimaging, where different problems may depend on different imaging modalities and where such preferences can help characterize modality contributions and guide the search for modality-specific biomarkers.

A related problem appears in multi-objective optimization. Rather than optimizing one fixed compromise between different objectives, Pareto Front Learning methods aim to learn a family of solutions parameterized by a preference vector. Pareto HyperNetworks (Navon et al., 2021), for example, generate models corresponding to different operating points, while other approaches directly condition a network on the desired preference (Ruchte & Grabocka, 2021). More recently, PaLoRA (Dimitriadis et al., 2025) showed that such a family can be represented efficiently using low-rank adapters, allowing the operating point to be chosen after training. These methods address trade-offs between *task objectives*. We instead aim to jointly pretrain a multimodal model f_θ, composed of

modality-specific encoders, that provides a family of preference-conditioned representations with different balances of shared and modality-specific information.

Building on this idea, we propose **STEER**, a preference-conditioned multimodal self-supervised learning framework that learns in a single training a family of representations within a single parameter-sharing model. STEER uses a preference vector

$$\boldsymbol{\lambda} = (\lambda_R, \lambda_{U_1}, \lambda_{U_2}) \in \Delta^2,$$

which controls the emphasis placed on redundant information and information specific to each modality. The preference acts both on the self-supervised training objective and through lightweight low-rank adaptations inside the encoder, steering the representation toward different information regimes. Instead of deciding during pretraining which regime future tasks will need, STEER learns in a single training several operating points across the simplex. For downstream tasks, the pretrained representation model remains fixed, and each task selects its preferred operating point $\boldsymbol{\lambda}^\star$.

We first evaluate STEER in a controlled setting where the shared and modality-specific factors are known by construction. This allows us to test whether moving through the preference space actually changes the information captured by the representation. STEER recovers the expected organization of shared(redundant) and unique factors and expands the attainable Pareto region beyond fixed-representation baselines. We then evaluate the framework on real multimodal tasks,where it outperforms fixed-representation baselines on several downstream problems while selecting distinct operating points from the same pretrained model. In neuroimaging, the selected preferences also agree with modality strengths identified independently of STEER, suggesting that the operating point provides an interpretable description of the multimodal information favored by a task.

Our contributions are as follows:

- We formulate multimodal SSL as a preference-conditioned trade-off between redundant and modality-specific information, allowing the information profile to be selected after pretraining.
- We introduce **STEER**, a single parameter-sharing model that learns a family of multimodal representations using simplex-valued preferences and low-rank adaptations.
- On a controlled benchmark, STEER recovers the expected shared–unique structure and expands the Pareto region beyond fixed-representation baselines.
- On real multimodal tasks, STEER performs best on several benchmarks while selecting task-specific operating points. In neuroimaging, it outperforms all baselines and recovers known modality strengths while remaining interpretable.

2 Related Work

Multimodal representation learning. Cross-modal contrastive methods such as CLIP (Radford et al., 2021) and GMC (Poklukar et al., 2022) mainly encourage agreement between modalities. FactorCL (Liang et al., 2023b) separates shared and modality-specific information, while CoMM (Dufumier et al., 2025) targets richer multimodal interactions including redundancy, uniqueness, and synergy. In all these cases, pretraining produces a fixed representation or fixed decomposition. STEER instead learns several shared–unique information profiles within the same pretrained model and lets the downstream task select among them.

Information decomposition. Partial Information Decomposition (PID) characterizes task-relevant multimodal information through redundancy, uniqueness, and synergy (Liang et al., 2023a). We use this view to define the axes explored by STEER, namely R, U_1, and U_2. We use the PID decomposition as a conceptual guide to define shared and modality-specific representation directions, rather than to estimate the exact PID values of the data.

Pareto front learning. Pareto Front Learning methods learn multiple trade-offs in a single model using a preference vector (Navon et al., 2021; Ruchte & Grabocka, 2021). PaLoRA (Dimitriadis et al., 2025) further shows that low-rank adapters can provide an efficient parameterization of such preference-conditioned solutions. These methods control trade-offs between task objectives. STEER uses the same general idea for a different purpose: controlling the balance between shared and modality-specific information inside a multimodal representation.

3 METHOD

Problem setup. Let $X_1, \ldots, X_n$ denote n modalities describing the same sample, and let $X = (X_1, \ldots, X_n)$. We aim to learn a self-supervised representation that can later support downstream tasks that are unknown during pretraining.

Instead of learning a single representation, we consider a family

$$Z(\boldsymbol{\lambda}) = f(X; \boldsymbol{\lambda}),$$

indexed by a preference vector

$$\boldsymbol{\lambda} = (\lambda_R, \lambda_{U_1}, \ldots, \lambda_{U_n}), \qquad \lambda_R + \sum_{m=1}^{n} \lambda_{U_m} = 1.$$

Here, λ_R controls the emphasis on information shared across modalities, while λ_{U_m} controls information specific to modality m.

In this work, we study the two-modality case,

$$\boldsymbol{\lambda} = (\lambda_R, \lambda_{U_1}, \lambda_{U_2}) \in \Delta^2.$$

The three vertices $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ correspond to the shared, modality-1-specific and modality-2-specific regimes, respectively. Interior points interpolate between them. For more than two modalities, additional partially shared and higher-order interactions may arise; we leave their explicit modeling for future work.

Our goal is therefore not to find one optimal preference during pretraining. Instead, we learn in a single training a model that exposes several information profiles and let the downstream task select the appropriate operating point afterward.

Redundant and modality-specific objectives. For modality m, the encoder produces

$$h_m = f_m(x_m; \boldsymbol{\lambda}),$$

which is followed by a redundancy head r_m and a modality-specific head u_m.

Shared information is encouraged through cross-modal InfoNCE (van den Oord et al., 2019):

$$\mathcal{L}_R = \text{InfoNCE}\left(r_1(h_1), r_2(h_2)\right).$$

For each modality, we preserve stable within-modality information through NT-Xent between two augmented views (Chen et al., 2020):

$$\mathcal{L}_{\text{self},m} = \text{NT-Xent}\left(u_m(h_m^a), u_m(h_m^b)\right).$$

This alone does not prevent the unique branch from encoding the same information as the redundant branch. Following the use of mutual-information upper bounds for shared–unique separation in FactorCL (Liang et al., 2023b), we use CLUB (Cheng et al., 2020) to reduce their dependence:

$$\widehat{I}_m = \text{CLUB}\left(u_m(h_m), \text{sg}[r_m(h_m)]\right),$$

giving

$$\mathcal{L}_{U_m} = \mathcal{L}_{\text{self},m} + \beta[\widehat{I}_m]_+.$$

For a preference $\boldsymbol{\lambda}$, the objective becomes

$$\mathcal{L}(\boldsymbol{\lambda}) = \lambda_R \mathcal{L}_R + \lambda_{U_1} \mathcal{L}_{U_1} + \lambda_{U_2} \mathcal{L}_{U_2}.$$

These objectives are guided by the redundancy–uniqueness view of PID (Liang et al., 2023a), but they are not estimators of the exact PID quantities. They simply encourage the representation toward shared or modality-specific information regimes.

Preference-conditioned encoder. Changing the loss weights alone can still lead to a common compromise across preferences. We therefore inject $\boldsymbol{\lambda}$ directly into the encoder using low-rank adaptations.

For a conditioned layer of modality m, we use

$$W_m(\boldsymbol{\lambda}) = W_{m,0} + \frac{\alpha}{r}\left(\lambda_R A_R B_R + \lambda_{U_m} A_{U_m} B_{U_m}\right),$$

where $W_{m,0}$ is the base weight, r the adapter rank, and α the adapter scaling factor. This follows the low-rank parameterization introduced by LoRA (Hu et al., 2021), and closely follows the preference-conditioned adapter construction of PaLoRA (Dimitriadis et al., 2025), adapted here to condition self-supervised modality encoders rather than task-specific supervised heads.

The preference now acts in two places: it weights the self-supervised objectives and changes the effective encoder parameters. Moving across the simplex therefore produces different representations rather than simply reweighting the same fixed features.

Preference training. STEER is trained over a discrete preference grid $\mathcal{G} \subset \Delta^2$. At each optimization step, a subset $\mathcal{G}_t \subseteq \mathcal{G}$ of M preferences is evaluated on the same minibatch and the corresponding losses are averaged before a single backward pass:

$$\mathcal{L}_{\text{STEER}} = \frac{1}{M} \sum_{\boldsymbol{\lambda} \in \mathcal{G}_t} \mathcal{L}(\widetilde{\boldsymbol{\lambda}}).$$

This allows several operating points to jointly update the same model while controlling the computational cost through M.

We adopt the preference-annealing transformation introduced in PaLoRA (Dimitriadis et al., 2025). For a target preference $\boldsymbol{\lambda}$,

$$\widetilde{\lambda}_k(\tau) = \frac{\lambda_k^{\gamma(\tau)}}{\sum_j \lambda_j^{\gamma(\tau)}}, \qquad \gamma(\tau) = \tau,$$

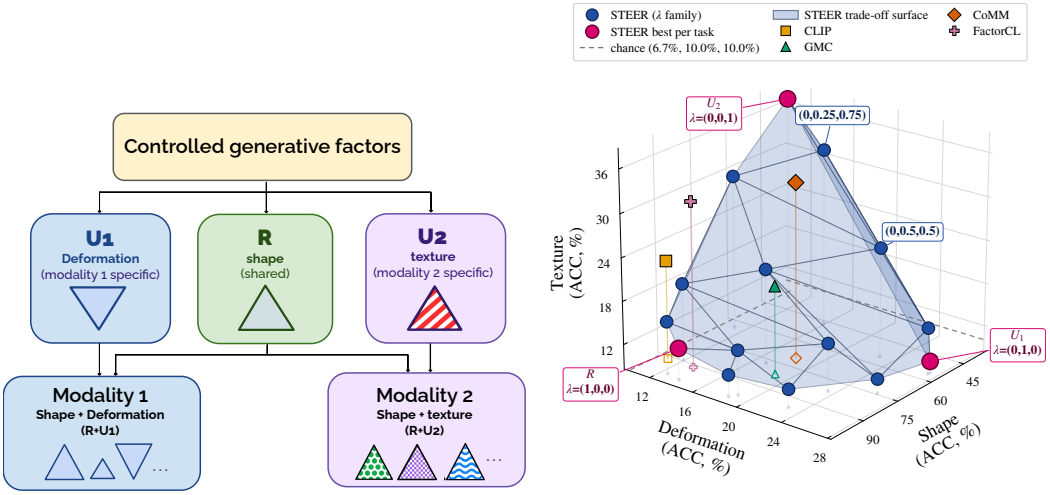
where $\tau \in [0, 1]$ denotes training progress. The transformation initially brings non-extreme preferences toward the center of the simplex face on which they lie and progressively recovers their nominal locations. In particular, the vertices remain unchanged throughout training, while intermediate preferences gradually separate toward their final operating points. Full details are provided in Appendix A.1.

Downstream preference selection. After pretraining, the representation model is frozen. For each downstream task and preference $\boldsymbol{\lambda}$, we construct a fixed-size representation $z(\boldsymbol{\lambda})$ from the redundant and modality-specific heads and train a lightweight probe on top of it. The exact read-out construction is given in Appendix A.3.

We evaluate all candidate preferences on the validation set. Since several nearby preferences often achieve very similar performance, choosing the exact validation maximum separately for each seed can make the selected operating point look less stable than it really is. We therefore identify a single consensus preference $\boldsymbol{\lambda}^\star$ across seeds and keep it fixed for the final test evaluation. The selection rule is described in Appendix A.4.

4 Experiments

Experimental Setup STEER is trained over a deterministic grid of preferences covering the simplex, with all preferences evaluated on the same minibatch before a single backward pass, as described in Section 3. Unless stated otherwise, we use the encoder-conditioned variant, where low-rank updates are inserted inside the encoder while the projection heads remain shared across preferences. At evaluation, the redundant and modality-specific blocks are combined according to $\boldsymbol{\lambda}$ using a fixed readout budget within each experiment. The grid resolution, annealing schedule, readout dimensions, and other implementation details are provided in Appendix A.1 and Appendix A.3.



(a) Controlled Trifeature construction.

(b) Learned shared–unique trade-off space.

Figure 2: Controlled validation on Trifeature. **(a)** The two modalities share shape R, while deformation U_1 and texture U_2 are modality-specific. **(b)** STEER operating points (blue and red circles) obtained by varying $\boldsymbol{\lambda}$, together with the fixed baselines. The STEER solutions define the Pareto front used for evaluation of the hypervolume (light-blue).

Evaluation Protocol All pretrained representations are frozen before downstream evaluation. Lightweight probes are trained for each operating point, and preference selection is performed only on the validation set. We report results over five independently trained seeds and use a single consensus preference $\boldsymbol{\lambda}^\star$ for each downstream task rather than a separate validation argmax for every seed. The selected preference is then fixed for test evaluation. The full consensus selection procedure is described in Appendix A.4.

4.1 Controlled Validation on Trifeature

To evaluate whether changing the preference $\boldsymbol{\lambda}$ actually changes the information contained in the learned representation, and whether STEER can recover different redundant and modality-specific information profiles within a single model, we design a synthetic bimodal dataset based on Trifeature (Hermann & Lampinen, 2020), following the controlled setting of CoMM (Dufumier et al., 2025). We pair samples from the two modalities such that shape is shared between them (R), while deformation is specific to modality 1 (U_1) and texture to modality 2 (U_2), allowing us to explicitly control the information available from each source. Following Hermann & Lampinen (2020) and CoMM, we use AlexNet (Krizhevsky et al., 2012) as modality-specific encoder, with embedding dimension $d = 256$. We also compare where preference-conditioned LoRA adapters should be applied, considering the encoder, the projection heads, or of them; the full placement ablation is reported in Appendix C.1.

Table 1: Linear evaluation top-1 accuracy (%) on the controlled Trifeature factors. Mean ± std over 5 seeds. The preference shown below each STEER score is the consensus operating point selected for that factor.

Method	Shape	Deformation	Texture
CLIP	87.95 ±13.78	18.56 ±1.86	18.07 ±2.63
GMC	74.60 ±1.44	25.71 ±1.04	16.73 ±1.19
CoMM	72.15 ±2.23	21.00 ±1.16	22.64 ±2.63
FactorCL	91.42 ±2.45	18.13 ±1.36	23.92 ±2.42
STEER	**99.98** ±0.05 $(1,0,0)$	**29.16** ±2.66 $(0,1,0)$	**37.46** ±3.96 $(0,0,1)$

This controlled experiment addresses three questions:

- **Does changing $\boldsymbol{\lambda}$ change the information contained in the representation?** Figure 2b shows that the different STEER operating points occupy distinct regions of the (R, U_1, U_2) space rather than

collapsing to a single solution. Moving across the simplex therefore changes which information is emphasized in the learned representation.

- **Can the same pretrained model specialize toward different information components?** Different operating points favor different factors, with top-1 accuracy reaching 99.98% for the shared factor R, 29.16% for U_1, and 37.46% for U_2. As reported in Table 1, the corresponding consensus preferences are the three expected vertices $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. These values exceed the best corresponding results of CLIP, GMC, CoMM, and FactorCL, while all three operating points are obtained from the same pretrained model.
- **Does STEER cover a richer set of information trade-offs than fixed representations?** We measure the volume dominated by the Pareto front in the (R, U_1, U_2) space. The joint front formed by the fixed baselines reaches a hypervolume of 0.0497, whereas the STEER family reaches 0.0591, corresponding to an 18.9% increase. STEER therefore covers with a single model and a single training a broader and more favorable region of the shared–unique trade-off space.

These results validate the main mechanism of STEER: a single pretrained model can expose genuinely different shared and modality-specific information profiles through $\boldsymbol{\lambda}$, rather than producing several equivalent representations.

4.2 Task-Specific Information Profiles on Real-World Benchmarks

We next evaluate STEER on real multimodal benchmarks, where the shared and modality-specific factors are no longer known by construction. We consider CMU-MOSEI (Bagher Zadeh et al., 2018) for non-neutral sentiment analysis, UR-FUNNY (Hasan et al., 2019) for humor detection, and AV-MNIST following the MultiBench setup (Liang et al., 2021) for audio-visual digit classification.The representation is pretrained without downstream labels, frozen for evaluation, and $\boldsymbol{\lambda}^\star$ is selected on validation data only.Ablations over M and the CLUB coefficient β are reported in Appendix C.2.

Table 2: Accuracy on real-world multimodal benchmarks. Results are reported as mean $\pm$ standard deviation across 5 seeds. STEER uses $M = 5$ preferences per minibatch, and the preference shown below each score is the consensus operating point selected using validation data.

Method	**CMU-MOSEI Accuracy (%)**	**UR-FUNNY Accuracy (%)**	**AV-MNIST Accuracy (%)**
CLIP	57.48 ± 1.63	54.63 ± 0.94	$\mathbf{71.12 \pm 0.40}$
GMC	69.83 ± 0.73	55.18 ± 0.51	64.06 ± 1.07
CoMM	68.48 ± 0.97	54.56 ± 1.62	61.05 ± 4.02
FactorCL	70.15 ± 0.64	56.86 ± 0.81	70.61 ± 0.39
STEER	$\mathbf{73.14 \pm 0.94}$ $(0, .25, .75)$	$\mathbf{58.03 \pm 1.06}$ $(0, .50, .50)$	69.70 ± 0.13 $(1, 0, 0)$

Table 2 shows that the selected information profile changes substantially across tasks. CMU-MOSEI favors the text modality-specific component, with $\boldsymbol{\lambda}^\star = (0, .25, .75)$, while UR-FUNNY selects an equal mixture of the two modality-specific components, $\boldsymbol{\lambda}^\star = (0, .5, .5)$. In contrast, AV-MNIST selects the pure shared vertex, $\boldsymbol{\lambda}^\star = (1, 0, 0)$.

STEER achieves the highest performance on CMU-MOSEI (73.14 ± 0.94) and UR-FUNNY (58.03 ± 1.57). On AV-MNIST, however, CLIP (71.12 ± 0.40) and FactorCL (70.61 ± 0.39) outperform STEER (69.70 ± 0.13). Interestingly, STEER itself selects the pure shared representation for this task. This provides a useful boundary case: the preference-conditioned family is most useful when downstream tasks benefit from different balances of shared and modality-specific information.

4.3 One Pretrained Model for Multiple Downstream Tasks

We next test whether a single pretrained STEER model can be reused across multiple downstream tasks with different information needs, while keeping the representation model fixed and selecting a task-specific operating point.

Table 3: Multi-task CMU-MOSEI results with STEER trained using $M = 5$ preferences per minibatch. The preference below each STEER score is the consensus operating point.

Method	Disgust Bal. Accuracy (%)	Happy Bal. Accuracy (%)	Sad Bal. Accuracy (%)	Average
CLIP	59.78 ± 3.15	54.64 ± 1.18	53.41 ± 1.18	55.94
GMC	65.71 ± 0.52	63.36 ± 1.11	58.79 ± 0.88	62.62
CoMM	66.00 ± 0.72	63.66 ± 0.86	58.98 ± 0.90	62.88
FactorCL	65.24 ± 1.15	61.72 ± 0.90	57.26 ± 0.64	61.41
STEER	$\mathbf{68.23 \pm 0.36}$ $(0, .25, .75)$	$\mathbf{63.87 \pm 1.01}$ $(.25, .50, .25)$	$\mathbf{59.16 \pm 0.78}$ $(.25, .25, .5)$	**63.75**

Multiple tasks on CMU-MOSEI. We evaluate a single model pretrained on CMU-MOSEI with $M = 5$ preferences per minibatch and reuse it for Disgust, Happy, and Sad prediction. The representation model remains fixed; only the downstream probe and selected preference change. As shown in Table 3, the three tasks select different operating points: $(0, .25, .75)$, $(.25, .25, .5)$, and $(.25, .5, .25)$, respectively. STEER achieves the best performance on all three tasks and the highest average accuracy. This shows that the tasks benefit from different information balances, which STEER can expose within the same pretrained model without representation retraining.

Application to multimodal neuroimaging. Neuroimaging provides a natural setting for the use case that motivated STEER. A multimodal model is typically pretrained once and later reused to study multiple pathologies, that rely on different imaging signals. We consider two complementary modalities extracted from the same cortical region: cortical folding morphology and structural connectivity derived from diffusion MRI. A full description of the modalities and the experiment is provided in the Appendix B.3. In our notation, U_1 denotes information specific to folding morphology, U_2 information specific to connectivity, and R information shared between the two. We keep the same pretrained STEER model frozen and evaluate it on three downstream problems with different expected modality requirements: prematurity classification, cognition score regression, and regression of a folding morphology shape index obtained from an Isomap embedding (Sun et al., 2012).

Table 4: Multimodal neuroimaging results. STEER uses the same pretrained representation family, trained with $M = 5$ preferences per minibatch, for all downstream tasks. Results are reported as mean $\pm$ standard deviation over the selected five runs. The preference shown below each STEER score is the consensus operating point selected for that task.

Method	Prematurity AUC	Cognition Pearson r	Isomap Pearson r
CLIP	0.621 ± 0.019	0.034 ± 0.003	0.439 ± 0.022
GMC	0.662 ± 0.019	0.051 ± 0.003	$\mathbf{0.716 \pm 0.004}$
CoMM	0.677 ± 0.011	0.050 ± 0.003	0.712 ± 0.004
FactorCL	0.652 ± 0.019	-0.024 ± 0.015	0.683 ± 0.012
STEER	$\mathbf{0.706 \pm 0.018}$ $(.75, .25, 0)$	$\mathbf{0.109 \pm 0.008}$ $(0, 0, 1)$	0.712 ± 0.002 $(0, 1, 0)$

These tasks provide a useful test of whether the selected operating point has a meaningful interpretation. Models trained separately on each modality indicate that prematurity can be predicted from both modalities, cognition is mainly associated with diffusion MRI, and the Isomap coordinates are primarily captured by folding morphology. The preferences selected by STEER follow the same pattern: prematurity selects a redundant heavy operating point $(.75, .25, 0)$, cognition selects the connectivity-specific vertex $(0, 0, 1)$, and Isomap selects the folding-specific vertex $(0, 1, 0)$.

With $M = 5$, STEER reaches an AUC of 0.706 for prematurity and a Pearson correlation of 0.109 for cognition, giving the best performance on these two tasks. On Isomap, STEER matches CoMM and remains close to GMC. The three tasks select substantially different information profiles from the same model, showing that STEER can adapt to different modality requirements through task-specific operating points while indicating the modality profile most relevant to each task.

Loss-weighted baseline control. A natural alternative to STEER is to vary the relative weights of the pretraining objectives in existing methods. We test this on CMU-MOSEI by training GMC, CoMM, and FactorCL over 15 different objective weightings and selecting the best configuration on validation. The weighting schemes are described in Appendix C.4.

Table 5 shows that simply sweeping the loss weights does not recover the same behavior. Even after training 15 separate models and selecting the best weighting on validation, the baselines remain below STEER on non-neutral sentiment accuracy. More importantly, each weighting corresponds to a different pretrained model. STEER instead learns these different information balances within a single parameter-sharing model, so the operating point can be chosen only after the downstream task is known. With $M = 5$, STEER also requires less total training time and substantially fewer stored parameters than the full weighted sweep of GMC, CoMM, or FactorCL.

Table 5: Loss-weighted baseline control on CMU-MOSEI. Each baseline searches over 15 independently pretrained models, while STEER ($M = 5$) learns all operating points within a single model.

Method	#Models	Acc.	Train-Time	Stored-parameters
GMC	15	70.69	2.05 h	26.3 M
CoMM	15	69.25	2.07 h	26.3 M
FactorCL	15	70.19	5.09 h	26.4 M
STEER	**1**	**73.14**	**1.71 h**	**2.09 M**

5 Discussion and Future Directions

Our experiments suggest that different multimodal tasks benefit from different balances of shared and modality-specific information. This is visible both in the controlled setting and on real data, where the selected preferences vary substantially across tasks. STEER therefore aims less at finding one universally best representation than at delaying the choice of information balance until the downstream task is known.

The AV-MNIST result also shows when this flexibility is less useful: the selected solution lies near the shared-information regime, where fixed cross-modal representations already perform strongly. In contrast, the MOSEI multi-task and neuroimaging experiments show the value of a single pretrained model when different downstream tasks prefer different operating points.

Several extensions are scheduled for future works. STEER currently models redundancy and modality-specific information, but not synergy. Extending the preference space to synergistic information would provide a richer decomposition, although a reliable self-supervised objective remains challenging. Moving beyond two modalities is another direction, where introducing partial redundancies and higher-order interactions is needed.

Finally, future work could improve both stages of the framework independently. On the training side, more efficient preference-sampling strategies could reduce the cost of learning the representation family. On the downstream side, improved operating-point selection methods could make it easier to identify the most suitable representation for a new task.

6 Conclusion

We introduced STEER, a preference-conditioned multimodal representation framework that learns a family of shared and modality-specific information profiles within a single pretrained model. Across controlled, benchmark, and neuroimaging experiments, STEER outperforms fixed-representation baselines on most downstream tasks while allowing different tasks to select substantially different operating points.

This property is particularly relevant for multimodal foundation models, which are pretrained before their future downstream uses are known. Rather than committing during pretraining to a single balance of multimodal information, STEER preserves a family of representations from which each downstream task can select the information profile that best matches its requirements.

AI USE STATEMENT

Generative AI tools were used to assist with writing and language refinement, assistance in identifying relevant literature, code development and debugging. All AI-assisted text, references, code, analyses, and research suggestions were reviewed and verified by the authors. The authors take full responsibility for the scientific content, results, and conclusions of this work.

ETHICS STATEMENT

The neuroimaging experiments use existing data from the UK Biobank, ABCD, and Human Connectome Project cohorts, accessed under their respective data-use and access procedures. No new human-subject data were collected as part of this work. The analyses use de-identified research data and were conducted in accordance with the conditions associated with access to each dataset. The proposed method is intended for representation learning and research analysis; the downstream neuroimaging experiments are not intended for clinical decision-making.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings, architectures, optimization parameters, preference sampling and selection procedures, and dataset-specific pipelines in the appendix. All reported results are evaluated over multiple training seeds, with the downstream operating point selected using validation data before final evaluation. An anonymized implementation of STEER and the scripts used to reproduce the experiments are available at https://anonymous.4open.science/r/Steer-0EC3/.

REFERENCES

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2236–2246, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1208.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying Unique Information. *Entropy*, 16(4):2161–2183, April 2014. ISSN 1099-4300. doi: 10.3390/e16042161.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607. PMLR, November 2020.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A Contrastive Log-ratio Upper Bound of Mutual Information. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1779–1788. PMLR, November 2020.

Nikos Dimitriadis, Pascal Frossard, and François Fleuret. Pareto Low-Rank Adapters: Efficient Multi-Task Learning with Preferences. In *International Conference on Learning Representations*, volume 2025, pp. 22323–22346, May 2025.

Benoit Dufumier, Javiera Castillo Navarro, Devis Tuia, and Jean-Philippe Thiran. What to align in multimodal contrastive learning? In *International Conference on Learning Representations*, volume 2025, pp. 5408–5432, May 2025.

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 Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2046–2056, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1211.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? Exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9995–10006. Curran Associates, Inc., 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, October 2021.

Ben Jeurissen, Jacques-Donald Tournier, Thijs Dhollander, Alan Connelly, and Jan Sijbers. Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. *NeuroImage*, 103:411–426, December 2014. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2014.07.061.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Julien Laval, Robin Guiavarch, Antoine Dufournet, Racim Menasria, Barthélémy Drabczuk, Cristobal Mendoza, Saeb Tounsi, Chikh Abdelghani Baroud, Merieme Bourenane, Vanessa Troiani, William Snyder, Marisa A. Patti, Mylène Moyal, Marion Plaze, Arnaud Cachia, Federica Santacroce, Giorgia Committeri, Claire Cury, Kevin De Matos, Olivier Colliot, Zhong Yi Sun, Clara Fischer, Vincent Frouin, Pietro Gori, Denis Rivière, Joël Chavas, and Jean-François Mangin. Representation learning of human cortical folding to reveal long lasting neurodevelopmental signatures, September 2026.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A. Lee, Yuke Zhu, Ruslan Salakhutdinov, and Louis-Philippe Morency. MultiBench: Multiscale Benchmarks for Multimodal Representation Learning. In *Advances in neural information processing systems*, volume 2021, pp. 1–20, December 2021.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, Russ R Salakhutdinov, and Louis-Philippe Morency. Quantifying & Modeling Multimodal Interactions: An Information Decomposition Framework. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27351–27393. Curran Associates, Inc., 2023a. doi: 10.52202/075280-1192.

Paul Pu Liang, Zihao Deng, Martin Q. Ma, James Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized Contrastive Learning: Going Beyond Multi-view Redundancy. In *Advances in Neural Information Processing Systems*, volume 36, pp. 32971–32998. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1430.

Abdullah Makkeh, Dirk Oliver Theis, and Raul Vicente. BROJA-2PID: A Robust Estimator for Bivariate Partial Information Decomposition. *Entropy*, 20(4), April 2018. ISSN 1099-4300. doi: 10.3390/e20040271.

Racim Menasria, Julien Laval, Denis Rivière, Jessica Dubois, Jean-François Mangin, and Joël Chavas. Right Collateral Sulcus Morphology Correlates With Receptive Communication in Preterm Born Adolescents and Neonates. In *2026 IEEE 23rd International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, April 2026. doi: 10.1109/ISBI61048.2026.11515837.

Aviv Navon, Aviv Shamsian, Ethan Fetaya, and Gal Chechik. Learning the Pareto Front with Hypernetworks. In *International Conference on Learning Representations*, 2021.

Petra Poklukar, Miguel Vasco, Hang Yin, Francisco S. Melo, Ana Paiva, and Danica Kragic. Geometric Multimodal Contrastive Representation Learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 17782–17800. PMLR, June 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, July 2021.

Michael Ruchte and Josif Grabocka. Scalable Pareto Front Approximation for Deep Multi-Objective Learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1306–1311, December 2021. doi: 10.1109/ICDM51629.2021.00162.

Robert E. Smith, Jacques-Donald Tournier, Fernando Calamante, and Alan Connelly. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 62(3):1924–1938, September 2012. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2012.06.005.

Robert E. Smith, Jacques-Donald Tournier, Fernando Calamante, and Alan Connelly. SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *NeuroImage*, 119:338–351, October 2015. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2015.06.092.

Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory Collins. UK biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine*, 12(3):e1001779, March 2015. ISSN 1549-1277. doi: 10.1371/journal.pmed.1001779.

Zhong Yi Sun, Stefan Klöppel, Denis Rivière, Matthieu Perrot, Richard Frackowiak, Hartwig Siebner, and Jean-François Mangin. The effect of handedness on the shape of the central sulcus. *NeuroImage*, 60(1):332–339, March 2012. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2011.12.050.

J.-D. Tournier, F. Calamante, and A. Connelly. Improved probabilistic streamlines tractography by 2nd order integration over fibre orientation distributions. *Proc. Intl. Soc. Mag. Reson. Med. (ISMRM)*, 18, 01 2010.

J-Donald Tournier, Robert Smith, David Raffelt, Rami Tabbara, Thijs Dhollander, Maximilian Pietsch, Daan Christiaens, Ben Jeurissen, Chun-Hung Yeh, and Alan Connelly. *MRtrix3*: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202: 116137, November 2019. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2019.116137.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv*, arXiv:1807.03748, January 2019. doi: 10.48550/arXiv.1807.03748.

David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E. J. Behrens, Essa Yacoub, and Kamil Ugurbil. The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80:62–79, October 2013. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2013.05.041.

Nora D Volkow, George F Koob, Robert T Croyle, Diana W Bianchi, Joshua A Gordon, Walter J Koroshetz, Eliseo J Pérez-Stable, William T Riley, Michele H Bloch, Kevin Conway, et al. The conception of the abcd study: From substance use to a broad nih collaboration. *Developmental cognitive neuroscience*, 32:4–7, 2018.

A Implementation and Evaluation Details

This supplementary material provides additional implementation details, ablations, controlled analyses, and dataset-specific experimental pipelines for STEER.

A.1 Preference Grid, Sampling, and Annealing

We define the preference space using a side-5 triangular lattice,

$$\mathcal{G} = \left\{ \boldsymbol{\lambda} \in \Delta^2 : 4\lambda_k \in \mathbb{Z} \right\},$$

which contains 15 candidate operating points with spacing $1/4$.

At optimization step t, the same minibatch is evaluated at a subset $\mathcal{G}_t \subseteq \mathcal{G}$ containing M preferences. Their losses are averaged before a single backward pass:

$$\mathcal{L}_{\text{STEER}} = \frac{1}{M} \sum_{\boldsymbol{\lambda} \in \mathcal{G}_t} \mathcal{L}\left(\widetilde{\boldsymbol{\lambda}}\right).$$

When $M = 15$, $\mathcal{G}_t = \mathcal{G}$ and the complete simplex is evaluated at every optimization step.

For the reported MultiBench experiments, we use $M = 5$. Since $M < 15$, preferences are selected using a balanced shuffled-cycle scheduler: the 15 grid points are randomly shuffled and visited in groups of M, so that all operating points receive approximately the same number of updates over training. The effect of changing M is studied in Appendix C.2.

For the reported neuroimaging experiments, we also use $M = 5$, but preferences are sampled independently from $\mathcal{G}$ at each minibatch rather than through the balanced cycle. Thus, the definition of the representation family is unchanged across experiments; only the number and sampling of preferences used for each optimization step differ.

Preference annealing. We use the power-based preference transformation introduced in PaLoRA (Dimitriadis et al., 2025):

$$\widetilde{\lambda}_k(\tau) = \frac{\lambda_k^{\gamma(\tau)}}{\sum_j \lambda_j^{\gamma(\tau)}}, \qquad \gamma(\tau) = \frac{\tau}{Q},$$

where $\tau \in [0, 1]$ denotes training progress and $Q = 1$ in all experiments. At the end of training, $\gamma(1) = 1$ and therefore $\widetilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$.

The effect of this transformation depends on the location of the preference on the simplex. Preferences with three non-zero coordinates are initially moved toward the simplex centroid and progressively separate toward their nominal locations. Preferences lying on an edge remain on that edge and move toward its midpoint, while a zero coordinate remains zero for any $\tau > 0$. Consequently, the three vertices are unchanged throughout training. The extreme objectives therefore remain available as anchors, while intermediate operating points progressively specialize toward their final preferences.

A.2 Optimization

The Trifeature models are trained for 100 epochs with AdamW, a base learning rate of 3×10^{-4}, weight decay 10^{-4}, and cosine learning-rate decay. LoRA parameters use a $10\times$ learning-rate multiplier. The CLUB critic is optimized separately with Adam at learning rate 3×10^{-4}. We use a batch size of 128 and temperature 0.1 for both InfoNCE and NT-Xent. Results are reported over five training seeds, 42–46.

Training views use `RandomResizedCrop(224, scale=(0.5,1.0))`, random horizontal flipping, color jitter, random grayscale, tensor conversion, and ImageNet normalization. Evaluation uses deterministic resizing and center cropping followed by ImageNet normalization.

A.3 Preference-Dependent Read-out

The downstream representation is assembled from three blocks:

$$\widehat{R} = \text{normalize}\left(\frac{r_1(h_1) + r_2(h_2)}{2}\right), \qquad U_1 = u_1(h_1), \qquad U_2 = u_2(h_2).$$

For the `dim` read-out used in the main results, a fixed representation budget D is divided across the three blocks according to the preference:

$$d_R + d_{U_1} + d_{U_2} = D, \qquad d_k \approx \lambda_k D.$$

Integer dimensions are assigned using largest-remainder allocation. The selected coordinates are concatenated and normalized:

$$z(\boldsymbol{\lambda}) = \left[\Pi_{d_R}(\widehat{R}); \Pi_{d_{U_1}}(U_1); \Pi_{d_{U_2}}(U_2)\right].$$

For Trifeature, $D = 256$. The total representation width is therefore identical at every operating point; only its allocation across information blocks changes.

Table 6: Dimension allocation for representative Trifeature preferences.

$\boldsymbol{\lambda}$	d_R	d_{U_1}	d_{U_2}	Total
$(1, 0, 0)$	256	0	0	256
$(0.5, 0.25, 0.25)$	128	64	64	256
$(0.25, 0.5, 0.25)$	64	128	64	256
$(0, 0, 1)$	0	0	256	256

For the MultiBench experiments, the corresponding read-out budget is $D = 64$. The representation width is again kept fixed across preferences within each experiment.

A.4 Consensus Operating-Point Selection

The validation response over the preference simplex can possibly be flat in some regions. Choosing an independent validation argmax for every training seed can therefore make nearly equivalent operating points appear artificially different. We use a noise-aware consensus rule to select one operating point per downstream task.

Let $a_s(\boldsymbol{\lambda})$ denote validation performance for seed s. We estimate the scale of seed-to-seed variation by

$$\varepsilon = \frac{1}{|\mathcal{G}|} \sum_{\boldsymbol{\lambda} \in \mathcal{G}} \mathrm{sd}_s \, a_s(\boldsymbol{\lambda}).$$

For seed s, preference $\boldsymbol{\lambda}$ is considered acceptable when its regret relative to that seed's best preference is no larger than this noise level:

$$\max_{\boldsymbol{\lambda}'} a_s(\boldsymbol{\lambda}') - a_s(\boldsymbol{\lambda}) \leq \varepsilon.$$

We first choose the preference accepted by the largest number of seeds. Ties are broken by the worst rank across seeds and then by mean rank. This yields a single consensus operating point $\boldsymbol{\lambda}^\star$ for each downstream task. Once selected, $\boldsymbol{\lambda}^\star$ is fixed before evaluating the held-out test set. The goal is to identify a reproducible near-optimal region rather than over-interpret small differences between neighboring preferences.

B Dataset-Specific Experimental Pipelines

B.1 Controlled Trifeature Setup

Dataset. We use the controlled Trifeature benchmark following Hermann & Lampinen (2020) and the multimodal construction of Dufumier et al. (2025). The dataset contains 2,800 records, split into 2,400 training, 200 validation, and 200 test examples. Images are generated at 64×64 resolution and resized to 224×224 before being passed to the encoder.

For the modality pair used in the main experiments, each sample is described by three independent latent factors:

$$R \in \{1, \ldots, 15\}, \qquad U_1 \in \{1, \ldots, 10\}, \qquad U_2 \in \{1, \ldots, 10\}.$$

The shared factor R determines shape. Modality 1 contains the shared shape together with a size-and-position deformation U_1, while modality 2 contains the same shape together with a texture factor U_2:

$$X_1 = (R, U_1), \qquad X_2 = (R, U_2).$$

This construction provides a controlled setting in which redundant and modality-specific information are known by design.

Deformation factor. The deformation variable U_1 does not change the identity of the underlying shape. Instead, it modifies its scale and position within the image. The ten deformation classes correspond to different combinations of scaling and translation while the base shape remains unchanged. We therefore refer to U_1 as a deformation factor throughout the paper.

Architecture. We use one AlexNet encoder per modality, following the implementation used by CoMM. The convolutional backbone produces $256 \times 6 \times 6 = 9{,}216$ features, followed by a linear projection to a 512-dimensional L_2-normalized representation. Each modality has one redundant head r_m and one modality-specific head u_m:

$$512 \to 512 \to 256,$$

with a ReLU between the two linear layers and L_2 normalization at the output.

For STEER, preference-conditioned low-rank adapters are applied to the encoder projection:

$$W_m(\boldsymbol{\lambda}) = W_{m,0} + \frac{\alpha}{r} \left(\lambda_R A_R B_R + \lambda_{U_m} A_{U_m} B_{U_m} \right),$$

where $r = 4$ and $\alpha = 8$ in the reported configuration. The LoRA matrices are initialized such that training starts from the unmodified base network (Hu et al., 2021).

B.2 MultiBench Experimental Setup

Datasets and modalities. We evaluate STEER on three MultiBench datasets: CMU-MOSEI (Bagher Zadeh et al., 2018), UR-FUNNY (Hasan et al., 2019), and AV-MNIST (Liang et al., 2021). CMU-MOSEI and UR-FUNNY use vision and text, whereas AV-MNIST pairs an image with an audio spectrogram. The dataset-specific inputs, splits, and downstream metrics are summarized in Table 7.

Table 7: MultiBench datasets and downstream evaluation settings.

	CMU-MOSEI	UR-FUNNY	AV-MNIST
Modality 1	Vision, 35-d/frame	Vision, 371-d/frame	Image, $1 \times 28 \times 28$
Modality 2	Text, 300-d/frame	Text, 300-d/frame	Audio, $1 \times 112 \times 112$
Sequence length	50	20	–
Task	Sentiment	Humour	Digit classification
Train / val. / test	Standard split	8074 / 1034 / 1058	55000 / 5000 / 10000
Metric	Accuracy	Accuracy	Accuracy

For CMU-MOSEI, sentiment scores lie in $[-3, 3]$ and are binarized by sign. We additionally report non-neutral Acc_2, where samples with sentiment exactly equal to zero are removed. UR-FUNNY is evaluated as binary humour classification. AV-MNIST follows the standard 10-class digit classification setting.

Architecture and preference conditioning. CMU-MOSEI and UR-FUNNY use one Transformer encoder per modality, following the CoMM implementation (Dufumier et al., 2025). Each encoder has width 40 and five attention heads. AV-MNIST instead uses the MultiBench LeNet image and audio backbones, whose outputs are projected to the same 40-dimensional encoder space. In all three datasets, each modality is followed by one redundant head r_m and one modality-specific head u_m, with 64-dimensional outputs.

The preference $\boldsymbol{\lambda} = (\lambda_R, \lambda_{U_1}, \lambda_{U_2})$ is defined on the same side-5 triangular grid of 15 operating points. Preference-conditioned rank-4 LoRA adapters are inserted in the encoders, while the four projection heads remain preference-independent. The adapter parameters use a $10\times$ learning-rate multiplier relative to the base network. Preferences are annealed from the simplex center toward their nominal locations during training.

The full-grid configuration evaluates all 15 preferences on every minibatch. Experiments using a reduced number M evaluate only M preferences at each step with the balanced shuffled-cycle scheduler described in Appendix C.2; the value of M used for a reported experiment is stated explicitly.

Training. STEER is optimized with AdamW using learning rate 3×10^{-4}, weight decay 10^{-4}, and batch size 128. CMU-MOSEI is trained for 200 epochs, while UR-FUNNY and AV-MNIST are trained for 100 epochs. Results are reported over five seeds, 42–46, with one H100 GPU per run.

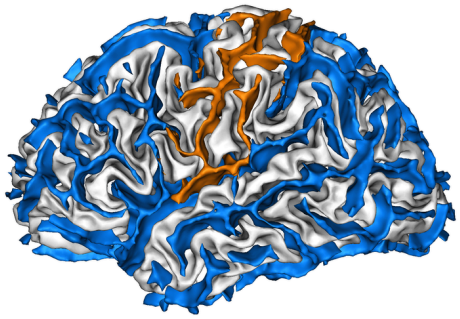
Read-out and downstream evaluation. For every preference, the downstream representation has a fixed width of 64. The `dim` read-out allocates this budget across the R, U_1, and U_2 blocks according to $\boldsymbol{\lambda}$, as described in Appendix A.3. Downstream evaluation uses logistic regression on standardized frozen representations. The regularization coefficient is selected from $\{10^{-2}, 10^{-1}, 1, 10\}$, with a wider grid $\{10^{-3}, \ldots, 10^{4}\}$ for the multi-task experiments. The probe is fitted on the validation split and evaluated on the test split. The task-specific operating point is selected with the consensus procedure described in Appendix A.4.

CLIP, GMC, CoMM, and FactorCL use the same input data and encoder front-end and are evaluated with the same downstream protocol. Their read-out dimensions are matched to STEER's 64-dimensional representation whenever possible; FactorCL uses the concatenation of its official heads, giving a 70-dimensional representation.

B.3 Neuroimaging Experimental setup

Modalities and pretrained encoders. We consider two complementary modalities extracted from the same left central sulcus cortical region: sulcal morphology derived from T1w MRI and tract endpoint density images derived from diffusion MRI tractography (Fig.3). For the sulcal morphology, the brains were first affinely aligned to a common template (the MNI space), allowing for the consistent cropping of a cortical area encompassing the central sulcus. Cortical folding is extracted from MRI as a 3D binary one-voxel-wide skeleton following the medial surface of sulci between the deepest sulcus points and the brain hull (Fig. 3a and 3b). The skeleton is then resampled at a 2 mm isotropic resolution (see Laval et al. (2026) for a detailed description of the modality). Diffusion MRI data from the UK Biobank Sudlow et al. (2015) were acquired at two b-values (1000 and 2000 s/mm^2), with 50 diffusion directions per shell, at an isotropic resolution of 2 mm. White matter fibre orientation distributions were estimated using multi-shell multi-tissue constrained spherical deconvolution (MSMT-CSD) Jeurissen et al. (2014). Whole-brain probabilistic tractography was then performed for 37,086 participants using MRtrix3 Tournier et al. (2019), with the second-order integration over fibre orientation distributions (iFOD2) algorithm Tournier et al. (2010) and anatomically constrained tractography (ACT) Smith et al. (2012), yielding 10 million streamlines per participant. First, the sulcal region mask was transformed from MNI space to diffusion space using the subject-specific affine transformation and nearest-neighbour interpolation. Streamlines with at least one endpoint inside the region were then extracted from the global tractogram using MRtrix3's `tckedit` command (Fig. 3c). Tract endpoint density images were generated at 1 mm isotropic resolution using `tckmap` with the `-ends_only` option, weighting each streamline by its SIFT2 weight Smith et al. (2015). Finally, the density images were affinely transformed to MNI space and cropped to the region's bounding box (Fig. 3d).

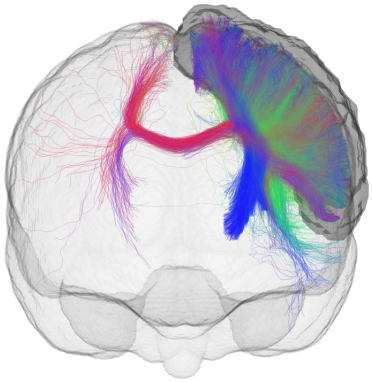
Sulcal morphology is encoded with Champollion Laval et al. (2026), while white matter features is encoded with ALMA (ALMA has the same architecture as the Champollion encoder, but with hyperparameters tuned for diffusion-related tasks). Both pretrained encoders remain frozen during STEER training.



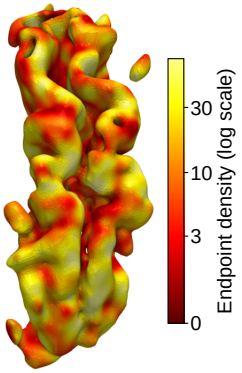
(a) Whole-brain cortical folding skeleton



(b) Champollion input crop



(c) Streamlines of the left central sulcus region



(d) ALMA input crop

Figure 3: Left central sulcus region of two UK Biobank subjects. (a) The cortical skeleton (orange and blue) is the object under study measuring the folds of the cortex. (b) Champollion input: the left central sulcus region crop of the cortical skeleton. (c) Streamlines extracted for the left central sulcus region, coloured by direction (red: left–right, green: anterior–posterior, blue: superior–inferior), with the region mask shown in grey. (d) ALMA input crop: streamline endpoint density, coloured by local endpoint density on a logarithmic scale. The isosurface renderings in (b) and (d) and the logarithmic colour scale in (d) are used for visualization only; the networks take the 3D crop volumes as input.

Table 8: Neuroimaging modalities and frozen pretrained encoders.

	Champollion	ALMA
Modality	Sulcal morphology	Tractography
Region	Left central sulcus	Left central sulcus
Input size	$1 \times 38 \times 36 \times 49$	$1 \times 80 \times 72 \times 104$
Embedding width	256	512

The multimodal training cohort contains 36,937 UK Biobank subjects (Sudlow et al., 2015) for whom both Champollion and ALMA inputs are available. Two augmented views are generated per subject and modality during training, while representation extraction uses deterministic preprocessing.

STEER adaptation. The neuroimaging preference is

$$\boldsymbol{\lambda} = (\lambda_R, \lambda_{U_C}, \lambda_{U_A}),$$

where U_C denotes information specific to sulcal morphology and U_A information specific to diffusion-derived white matter features. The Champollion encoder receives $(\lambda_R, \lambda_{U_C})$, whereas ALMA receives $(\lambda_R, \lambda_{U_A})$. Each modality has one redundant and one modality-specific projection head, r_C, u_C, r_A, u_A, with hidden width 128 and output width 64.

Adapters in the last-stage regime. The pretrained Champollion and ALMA encoders remain frozen, and preference conditioning is introduced through low-rank adapters placed only in the last stage of each encoder. Concretely, we adapt the convolutions of the final down-sampling block and the final linear layer, while all earlier layers remain unchanged. Each adapted layer contains one redundant and one modality-specific branch, mixed according to the preference received by that encoder. We use a rank ratio of 0.15, with adapter scales $\alpha_C = 4$ for Champollion and $\alpha_A = 0.25$ for ALMA. A single set of adapter parameters is shared across all preferences, so changing $\boldsymbol{\lambda}$ changes the representation without requiring a separate model for each operating point.

Training. Training uses the same redundant and modality-specific objectives introduced in the main paper: cross-modal InfoNCE for the redundant branch, and within-modality NT-Xent together with the CLUB dependence penalty for the modality-specific branches. The preference space contains the same 15 simplex points used in the other experiments. In the reported neuroimaging configuration, $M = 5$ preferences are sampled per optimization step and their losses are averaged.

Models are trained for 20 epochs with AdamW, learning rate 10^{-3}, weight decay 0.01, batch size 16, and CLUB coefficient $\beta = 1.0$. The two CLUB critics are optimized separately with learning rate 10^{-3}.

Read-out and downstream tasks. The downstream representation uses the same dimension-allocation principle as the other experiments, with a fixed width of 64 for every preference. After multimodal training, STEER is frozen and evaluated on cohorts different from the UK Biobank training cohort.

Table 9: Neuroimaging downstream tasks and evaluation protocols.

Task	Cohort	n	Probe	Metric
Prematurity	ABCD	4236	Logistic regression	AUC
Cognition	HCP	730	Ridge regression	Pearson r
Sulcal morphology	HCP	651	Ridge regression	Pearson r

Prematurity distinguishes subjects born at 28–32 weeks gestational age from those born at 37 weeks or later, based on subjects from the ABCD cohort (Volkow et al., 2018). Age, sex, site, and household socioeconomic status are residualized within each fold, and evaluation uses five-fold cross-validation with class-balanced logistic regression as described in Menasria et al. (2026). Cognition predicts the Human Connectome Project (HCP, Van Essen et al. (2013)) `varimax_cog` score after removing age and sex from the target, using RidgeCV over repeated ten-fold splits. The sulcal morphology task predicts six left central sulcus Isomap coordinates from Sun et al. (2012) using repeated ten-fold cross-validation and RidgeCV.

For each task, the consensus preference is computed across training seeds from the corresponding cross-validated downstream scores. The reported score is the mean and standard deviation across seeds at that single consensus operating point on a test set.

The neuroimaging baselines use the same frozen Champollion and ALMA encoders, the same downstream representation width whenever possible, and the same probe functions.

C Ablation Studies

C.1 Where Should the Preference Act?

STEER introduces preference conditioning through low-rank adaptations, but the conditioning can be inserted at different locations in the representation pipeline. We compare three variants: conditioning the encoder only (`enc`), conditioning the projection heads only (`proj`), and conditioning both (`both`). All other components are kept fixed.

Figure 4 illustrates the reported `enc` configuration. For modality 1, the encoder is conditioned by $(\lambda_R, \lambda_{U_1})$, while modality 2 receives $(\lambda_R, \lambda_{U_2})$. The preference-conditioned LoRA weights therefore act directly inside each encoder and produce $h_1(\boldsymbol{\lambda})$ and $h_2(\boldsymbol{\lambda})$. The redundant and unique projection heads, r_m and u_m, remain shared across preferences. Thus, changing $\boldsymbol{\lambda}$ modifies the representation produced by the encoder itself rather than only changing its final projection.

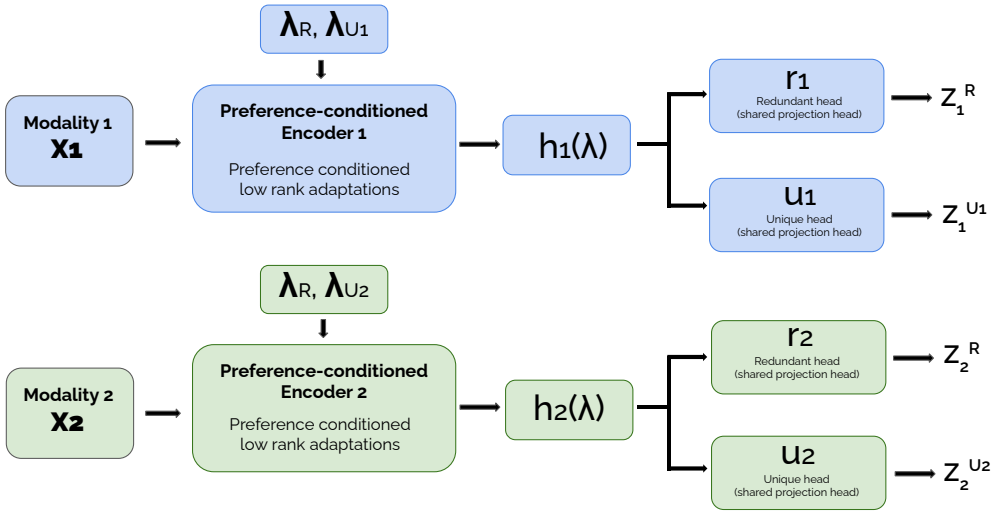


Figure 4: Encoder-conditioned STEER architecture. Preference-conditioned low-rank adaptations are applied inside each modality encoder. Modality 1 is conditioned by $(\lambda_R, \lambda_{U_1})$ and modality 2 by $(\lambda_R, \lambda_{U_2})$. The redundant heads r_1, r_2 and unique heads u_1, u_2 are preference-independent and shared across all operating points.

To determine whether conditioning the encoder is preferable to conditioning later stages of the network, we move the same preference-conditioned adapters to the projection heads (`proj`) or place them in both the encoder and projection heads (`Full`). Adapter strength is controlled through $\alpha \in \{2, 4, 8\}$. Each value is first evaluated on the validation split using three seeds, after which the selected value is evaluated on the test split with five seeds. $\alpha = 8$ is selected for all three placements.

Pareto metrics. All Pareto metrics are computed from probe performance on the same held-out Trifeature test examples for STEER and the fixed baselines. Classification accuracies are expressed as fractions in $[0, 1]$, with no additional normalization. Hypervolume is computed with the origin as reference point. In the main Trifeature analysis, we use the full three-objective space (R, U_1, U_2), yielding hypervolumes of 0.0497 for the joint baseline front and 0.0591 for STEER. In the placement ablation below, we report the learned-family hypervolume HV_λ in the two-dimensional space (R, U_{avg}), where $U_{\text{avg}} = (U_1 + U_2)/2$. Pareto Coverage Gain (PCG) denotes the additional hypervolume contributed beyond the fixed baseline front:

$$\text{PCG} = HV(\mathcal{F}_{\text{base}} \cup \mathcal{F}_{\text{STEER}}) - HV(\mathcal{F}_{\text{base}}).$$

The encoder-conditioned variant gives the strongest overall result, with the highest HV_λ and the highest average performance across the three controlled factors. Conditioning only the projection

Table 10: Preference-conditioning placement on Trifeature. Results are reported on the test split over five seeds after selecting α on validation.HV_λ denotes the two-dimensional hypervolume of the learned preference family. Average is computed over Shape, Deformation, and Texture.

Placement	α	HV_λ	Shape	Deformation	Texture	Avg.
Encoder	8	$\mathbf{0.2198 \pm 0.0077}$	$\mathbf{0.9998}$	$\mathbf{0.2728}$	0.3714	$\mathbf{0.5480}$
Projection heads	8	0.2138 ± 0.0103	0.9948	0.2010	0.3749	0.5236
Full	8	0.2029 ± 0.0095	0.9969	0.1829	$\mathbf{0.3909}$	0.5236

heads reaches a similar shared-factor accuracy but is substantially weaker on deformation, while conditioning both locations does not provide an additional benefit. These results motivate the encoder-conditioned design used throughout the main experiments: the preference acts directly on the learned representation, while the downstream projection heads remain common to the whole representation family.

C.2 Preference Count and Dependence-Penalty Strength

STEER has two training choices that can affect the learned representation family. The number of preferences M evaluated per minibatch controls how densely the preference simplex is sampled at each optimization step, while β controls the strength of the CLUB dependence penalty between redundant and modality-specific branches. We study the two choices jointly.

For $M < 15$, we use a balanced shuffled-cycle scheduler: only M preferences are evaluated on each minibatch, while the full simplex is still covered over training. We compare $M \in \{15, 5, 3\}$ and vary β over the values shown in Table 11.

Table 11: Joint ablation of the CLUB coefficient β and the number of preferences M evaluated per minibatch. Trifeature reports the learned-family hypervolume HV_λ, while CMU-MOSEI, UR-FUNNY, and AV-MNIST report downstream accuracy (%). Values are mean $\pm$ standard deviation over five seeds.

Dataset	β	$M = 15$	$M = 5$	$M = 3$
Trifeature (HV_λ)	0.5	0.2198 ± 0.0077	$\mathbf{0.2328 \pm 0.0120}$	0.2249 ± 0.0089
CMU-MOSEI	0.25	72.81 ± 0.91	$\mathbf{73.14 \pm 0.94}$	72.86 ± 0.87
	0.5	72.05 ± 1.19	72.73 ± 0.50	72.32 ± 0.95
	1.0	64.59 ± 2.78	69.45 ± 3.19	67.51 ± 2.85
UR-FUNNY	0.25	$\mathbf{59.51 \pm 0.83}$	58.03 ± 1.06	59.37 ± 0.82
	0.5	57.60 ± 1.73	57.82 ± 1.38	58.22 ± 0.75
	1.0	58.19 ± 1.76	58.13 ± 1.45	58.06 ± 2.12
AV-MNIST	0.25	69.45 ± 0.50	$\mathbf{69.70 \pm 0.13}$	69.68 ± 0.54
	0.5	69.16 ± 0.53	69.16 ± 0.46	69.41 ± 0.69
	1.0	68.30 ± 0.90	69.24 ± 0.55	69.15 ± 0.69

The results show that M can be reduced substantially without a systematic loss in downstream performance. On CMU-MOSEI, for example, accuracy changes from 72.81 ± 0.91 at $M = 15$ to 73.14 ± 0.94 at $M = 5$ for $\beta = 0.25$. A similar pattern is observed on AV-MNIST, while UR-FUNNY shows only modest variation across the tested preference counts. This suggests that evaluating the full simplex on every minibatch is not necessary to learn a useful representation family.

The effect of β is more dataset-dependent. On CMU-MOSEI, a large dependence penalty substantially reduces performance, whereas the differences are smaller on UR-FUNNY and AV-MNIST. We therefore treat β as a dataset-dependent hyperparameter rather than fixing a single value across all settings.

On the controlled Trifeature experiment, reducing M shows the same overall behavior: HV_λ is 0.2198 ± 0.0077, 0.2328 ± 0.0120, and 0.2249 ± 0.0089 for $M = 15$, $M = 5$, and $M = 3$, respectively. Together with the substantial reduction in training cost, these results show that $M = 5$ provides a practical computational saving while preserving the quality of the learned representation family.

C.3 Dependence-Penalty Strength

We vary the coefficient β multiplying the CLUB dependence penalty. The appropriate strength is dataset dependent, so this coefficient is treated as a validation hyperparameter rather than fixed universally.

Table 12: Effect of the CLUB coefficient β. Each row reports the downstream metric used for that dataset (Acc_2 for CMU-MOSEI and UR-FUNNY, top-1 for AV-MNIST), mean $\pm$ standard deviation over five seeds.Results are reported on the validation set. Bold is the best cell in each row; † marks the value used for the main results.

Dataset	$\beta = 0.25$	$\beta = 0.5$	$\beta = 1.0$
CMU-MOSEI	$\mathbf{65.83 \pm 1.34}^\dagger$	63.90 ± 2.49	59.79 ± 1.45
UR-FUNNY	$\mathbf{59.77 \pm 1.99}^\dagger$	58.56 ± 0.52	58.09 ± 0.78
AV-MNIST	$\mathbf{69.43 \pm 0.45}^\dagger$	69.16 ± 0.53	68.24 ± 0.94

The sweep confirms that the dependence penalty changes the learned family.

C.4 Loss-Weighted Baseline Control

The main paper compares STEER with loss-weighted versions of GMC, CoMM, and FactorCL. Here we describe how these weighted baselines are constructed. The weights are applied during *pretraining*; they do not reweight the representation or the downstream probe. For each baseline, we train 15 independent models with identical data, architecture, and optimization settings, changing only the relative contribution of the loss terms.

GMC and CoMM. Both GMC and CoMM contain two modality-to-joint objectives, associated with the visual and text modalities. We introduce weights (w_v, w_t) directly on these terms:

$$\mathcal{L}_{\text{GMC/CoMM}} = w_v \, \mathcal{L}_{v \to j} + w_t \, \mathcal{L}_{t \to j}, \qquad w_v + w_t = 1.$$

We evaluate

$$w_v \in \left\{ 0, \frac{1}{14}, \frac{2}{14}, \ldots, 1 \right\}, \qquad w_t = 1 - w_v,$$

which gives 15 independently pretrained models for each method. For example, the selected GMC weighting $0.214/0.786$ assigns approximately 21.4% of the weighted objective to the visual-to-joint term and 78.6% to the text-to-joint term.

FactorCL. FactorCL contains several contrastive terms rather than a two-term objective. To obtain an analogous three-way control, we group its six loss terms according to whether they contribute primarily to redundant, visual-specific, or text-specific information. In our implementation, the `r` and `ccl` terms receive the redundant weight w_R, `u_v` receives w_{U_1}, and `u_t` receives w_{U_2}. The cross terms `cl` and `cr` contribute to both unique branches and are weighted by

$$\frac{w_{U_1} + w_{U_2}}{2}.$$

The resulting objective is

$$\mathcal{L}_{\text{FactorCL}} = w_R\big(\mathcal{L}_r + \mathcal{L}_{ccl}\big) + w_{U_1}\mathcal{L}_{u_v} + w_{U_2}\mathcal{L}_{u_t} + \frac{w_{U_1} + w_{U_2}}{2}\big(\mathcal{L}_{cl} + \mathcal{L}_{cr}\big),$$

with

$$(w_R, w_{U_1}, w_{U_2}) \in \mathcal{G},$$

where $\mathcal{G}$ is the same 15-point simplex used by STEER. This grouping is specific to our control experiment and is not part of the original FactorCL formulation.

For all baselines, the weighting is selected using validation data only, and the held-out test set is evaluated once after selection. The selected configurations are reported below; the compact performance comparison is given in the main paper.

Table 13: Validation-selected configurations for the loss-weighted baseline control on CMU-MOSEI.

Method	Selected weights	Models	Val. Acc.
STEER ($M = 5$)	$(0, .25, .75)$	**1**	**67.78**
GMC	$0.214/0.786$	15	66.24
CoMM	$0.071/0.929$	15	64.90
FactorCL	$(0.25, 0, 0.75)$	15	61.96

D Information Decomposition Across the Preference Simplex

The controlled Trifeature experiment allows us to test more directly whether changing the preference modifies the type of information captured by the model. Linear-probe accuracy shows whether shape, deformation, and texture remain predictable, but it does not tell us whether these predictions rely on redundant or modality-specific information. We therefore complement the probe analysis with a model-level Partial Information Decomposition (PID) (Liang et al., 2023a) using a BROJA-style bivariate decomposition (Bertschinger et al., 2014; Makkeh et al., 2018).

Recall that Trifeature is constructed as

$$X_1 = (R, U_1), \qquad X_2 = (R, U_2),$$

where shape R is shared by the two modalities, deformation U_1 is available only in modality 1, and texture U_2 only in modality 2. For each preference $\boldsymbol{\lambda}$, we use the model prediction for each controlled factor as the PID target. Given the two modalities as sources and a model prediction $\widehat{Y}$ as target, BROJA decomposes their information as

$$I(X_1, X_2; \widehat{Y}) = R + U_1 + U_2 + S,$$

where R denotes redundant information, U_1 and U_2 information unique to each modality, and S synergistic information.

We apply this decomposition separately to the predictions of the three controlled factors. For shape, which is shared by construction, we report the redundant component. For deformation, we report the information unique to modality 1, and for texture the information unique to modality 2. Synergy is not reported here because the present analysis focuses on the three information components explicitly controlled by STEER. To reduce finite-sample bias, we repeat the decomposition after shuffling the prediction target and subtract the resulting value as a finite-sample floor.

Table 14: Model-level PID at the three vertices of the Trifeature preference simplex. Each column reports the PID component associated with its controlled factor: redundant information for shape, modality-1 unique information for deformation, and modality-2 unique information for texture. Values are in bits and reported as mean $\pm$ standard deviation. Bold indicates the information component targeted by each preference vertex.

Preference	Shape (R)	Deformation (U_1)	Texture (U_2)
$(1, 0, 0)$	$\mathbf{3.886 \pm 0.012}$	0.029 ± 0.003	0.099 ± 0.012
$(0, 1, 0)$	1.448 ± 0.041	$\mathbf{1.111 \pm 0.067}$	0.002 ± 0.001
$(0, 0, 1)$	0.942 ± 0.087	0.000 ± 0.002	$\mathbf{1.317 \pm 0.078}$

The three vertices already show the expected change in information profile: the shared-information preference produces the largest redundant contribution, while the two modality-specific preferences increase the unique information associated with their respective modalities. We then extend the analysis to all 15 operating points of the simplex. Each measured information component increases strongly with its corresponding preference coordinate:

$$\rho(\lambda_R, R) = 0.968, \qquad \rho(\lambda_{U_1}, U_1) = 0.968, \qquad \rho(\lambda_{U_2}, U_2) = 0.952,$$

where ρ denotes Spearman correlation across the preference simplex.

This gives a stronger controlled validation than probe accuracy alone. Changing $\boldsymbol{\lambda}$ does not simply change how well a downstream factor can be predicted; it moves the model through measurably different redundant and modality-specific information regimes. Since the underlying information structure of Trifeature is known by construction, this experiment provides a direct check that the STEER preference axes behave as intended.