Does my multimodal model learn cross-modal interactions? It's harder to tell than you might think!

Jack Hessel
Allen Institute for AI
jackh@allenai.org

Lillian Lee
Cornell University
llee@cs.cornell.edu

Abstract

Modeling expressive cross-modal interactions seems crucial in multimodal tasks, such as visual question answering. However, sometimes high-performing black-box algorithms turn out to be mostly exploiting unimodal signals in the data. We propose a new diagnostic tool, *empirical multimodally-additive function projection* (EMAP), for isolating whether or not cross-modal interactions improve performance for a given model on a given task. This function projection modifies model predictions so that cross-modal interactions are eliminated, isolating the additive, unimodal structure. For seven image+text classification tasks (on each of which we set new state-of-the-art benchmarks), we find that, in many cases, removing cross-modal interactions results in little to no performance degradation. Surprisingly, this holds even when expressive models, with capacity to consider interactions, otherwise outperform less expressive models; thus, performance improvements, even when present, often cannot be attributed to consideration of cross-modal feature interactions. We hence recommend that researchers in multimodal machine learning report the performance not only of unimodal baselines, but also the EMAP of their best-performing model.

1 Introduction

Given the presumed importance of reasoning across modalities in multimodal machine learning tasks, we should evaluate a model's ability to leverage cross-modal interactions. But such evaluation is not straightforward; for example, an early Visual Question-Answering (VQA) challenge was later "broken" by a high-performing method that ignored the image entirely (Jabri et al., 2016).

One response is to create multimodal-reasoning datasets that are specifically and cleverly balanced to resist language-only or visual-only models; examples are VQA 2.0 (Goyal et al., 2017), NLVR2 (Suhr et al., 2019), and GQA (Hudson and Manning, 2019). However, a balancing approach not always desirable. For example, if image+text data is collected from an online social network (such as for popularity prediction or sentiment analysis), post-hoc rebalancing may obscure trends in the original data-generating processs. So, what alternative diagnostic tools are available for better understanding what models learn?

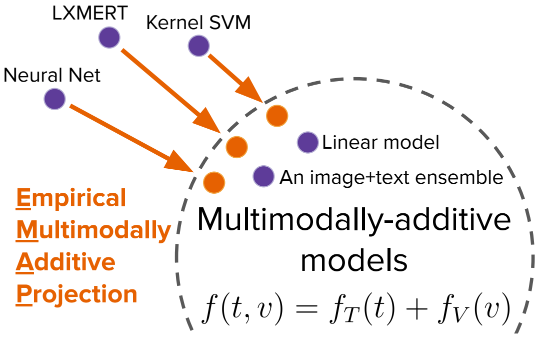
LXMERT
Kernel SVM
Neural Net
Linear model
An image+text ensemble
Empirical Multimodally Additive Projection
Multimodally-additive models
$f(t, v) = f_T(t) + f_V(v)$

Figure 1: We introduce EMAP, a diagnostic for classifiers that take in textual and visual inputs. Given a (black-box) trained model, EMAP computes the predictions of an image/text ensemble that best approximates the full model predictions via empirical function projection. Although the projected predictions lose visual-textual interaction signals exploited by the original model, they often perform suprisingly well.

The main tool utilized by prior work is *model comparison.* In addition to comparing against text-only and image-only baselines, often, two multimodal models with differing representational capacity (e.g., a cross-modal attentional neural network vs. a linear model) are trained and their performance compared. The argument commonly made is that if model A, with greater expressive capacity, outperforms model B, then the performance differences can be at least partially attributed to that increased expressivity.

But is that a reliable argument? Model perfor-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 861–877,
November 16–20, 2020. ©2020 Association for Computational Linguistics

mance comparisons are an opaque tool for analysis, especially for deep neural networks: performance differences versus baselines, frequently small in magnitude, can often be attributed to hyperparameter search schemes, random seeds, the number of models compared, etc. (Yogatama and Smith, 2015; Dodge et al., 2019). Thus, while model comparisons are an acceptable starting point for demonstrating whether or not a model is learning an interesting set of (or any!) cross-modal factors, they provide rather indirect evidence.

We propose *Empirical Multimodally-Additive*[1] *function Projection* (EMAP) as an additional diagnostic for analyzing multimodal classification models. Instead of comparing two different models, a single multimodal classifier's predictions are *projected* onto a less-expressive space: the result is equivalent to a set of predictions made by the closest possible ensemble of text-only and visual-only classifiers. The projection process is computationally efficient, has no hyperparameters to tune, can be implemented in a few lines of code, is provably unique and optimal, and works on any multimodal classifier: we apply it to models ranging from polynomial kernel SVMs to deep, pretrained, Transformer-based self-attention models.

We first verify that EMAPs do degrade performance for synthetic cases and for visual question answering cases where datasets have been specifically designed to require cross-modal reasoning. But we then examine a test suite of several recently-proposed multimodal prediction tasks that have *not* been specifically balanced in this way. We first achieve state-of-the-art performance for all of the datasets using a linear model. Next, we examine more expressive interactive models, e.g., pretrained Transformers, capable of cross-modal attention. While these models sometimes outperform the linear baseline, EMAP reveals that performance gains are (usually) not due to multimodal interactions being leveraged.

Takeaways: For future work on multimodal classification tasks, we recommend authors report the performance of: 1) unimodal baselines; 2) any multimodal models they consider; and, *critically,* 3) the *empirical multimodally-additive projection* (EMAP) of their best performing multimodal model (see §6 for our full recommendations).

2 Related Work

Constructed multimodal classification tasks. In addition to image question answering/reasoning datasets already mentioned in §1, other multimodal tasks have been constructed, e.g., video QA (Lei et al., 2018; Zellers et al., 2019), visual entailment (Xie et al., 2018), hateful multimodal meme detection (Kiela et al., 2020), and tasks related to visual dialog (de Vries et al., 2017). In these cases, unimodal baselines are shown to achieve lower performance relative to their expressive multimodal counterparts.

Collected multimodal corpora. Recent computational work has examined diverse multimodal corpora collected from in-vivo social processes, e.g., visual/textual advertisements (Hussain et al., 2017; Ye and Kovashka, 2018; Zhang et al., 2018), images with non-literal captions in news articles (Weiland et al., 2018), and image/text instructions in cooking how-to documents (Alikhani et al., 2019). In these cases, multimodal classification tasks are often proposed over these corpora as a means of testing different theories from semiotics (Barthes, 1988; O'Toole, 1994; Lemke, 1998; O'Halloran, 2004, inter alia); unlike many VQA-style datasets, they are generally not specifically balanced to force models to learn cross-modal interactions.

Without rebalancing, should we expect cross-modal interactions to be useful for these multimodal communication corpora? Some semioticians posit: *yes! Meaning multiplication* (Barthes, 1988) between images and text suggests, as summarized by Bateman (2014):

> under the right conditions, the value of a combination of different modes of meaning can be worth more than the information (whatever that might be) that we get from the modes when used alone. In other words, text 'multiplied by' images is more than text simply occurring with or alongside images.

Jones et al. (1979) provide experimental evidence of conditional, compositional interactions between image and text in a humor setting, concluding that "it is the dynamic interplay between picture and caption that describes the multiplicative relationship" between modalities. Taxonomies of the specific types of compositional relationships image-text pairs can exhibit have been proposed (Marsh and Domas White, 2003; Martinec and Salway, 2005).

[1] In §3, we more formally introduce *additivity*.

Model Interpretability. In contrast to methods that design more interpretable algorithms for prediction (Lakkaraju et al., 2016; Ustun and Rudin, 2016), several researchers aim to explain the behavior of complex, black-box predictors on individual instances (Štrumbelj and Kononenko, 2010; Ribeiro et al., 2016). The most related of these methods to the present work is Ribeiro et al. (2018), who search for small sets of "anchor" features that, when fixed, largely determine a model's output prediction on the input points. While similar in spirit to ours and other methods that "control for" a fixed subset of features (e.g., Breiman (2001)), their work 1) focuses only on high-precision, local explanations on single instances; 2) doesn't consider multimodal models (wherein feature interactions are combinatorially more challenging in comparison to unimodal models); and 3) wouldn't guarantee consideration of multimodal "anchors."

3 EMAP

Background. We consider models f that assign scores to textual-visual pairs (t, v), where t is a piece of text (e.g., a sentence), and v is an image.[2] In multi-class classification settings, values $f(t, v) \in \mathbb{R}^d$ typically serve as per-class scores. In ranking settings $f(t_1, v_1)$ may be compared to $f(t_2, v_2)$ via a ranking objective.

We are particularly interested in the types of compositionality that f uses over its visual and textual inputs to produce scores. Specifically, we distinguish between *additive* models (Hastie and Tibshirani, 1987) and *interactive* models (Friedman, 2001; Friedman and Popescu, 2008).[3] A function f of a representation z of n input features is *additive* in I if it decomposes as

$$f(z) = f_I(z_I) + f_{\setminus I}(z_{\setminus I}),$$

where (setting $\mathcal{I} = \{1, \dots, n\}$ for convenience) $I \subset \mathcal{I}$ indexes a subset of the features, $\setminus I = \mathcal{I} \setminus I$, and for any $S \subset \mathcal{I}$, z_S is the restriction of z to only those features whose indices appear in S.

In our case, because features are the union of $\underline{T}$extual and $\underline{V}$isual predictors, we say that a model is *multimodally additive* if it decomposes as

$$f(t, v) = f_T(t) + f_V(v) \qquad (1)$$

Additive multimodal models are simply ensembles of unimodal classifiers and as such, may be considered underwhelming to multiple communities. A recent ACL paper, for example, refers to such ensembles as "naive." On the semiotics side, the conditionality implied by meaning multiplication (Barthes, 1988) — that the joint semantics of an image/caption depends non-linearly on its accompaniment — cannot be modeled additively: multimodally additive models posit that each image, independent of its text pairing, contributes a fixed score to per-class logits (and vice versa).

In contrast, *multimodally interactive* models are the set of functions that *cannot* be decomposed as in Equation 1 — that is, f's output conditionally depends on its inputs in a non-additive fashion.

Machine learning models. One canonical multimodally additive model is a linear model trained over a concatenation of textual and visual features $[t; v]$, i.e.,

$$f(t, v) = w^T[t; v] + b = \underbrace{w_t^T t}_{f_T(t)} + \underbrace{w_v^T v + b}_{f_V(v)}. \qquad (2)$$

We later detail several multimodally interactive models, including multi-layer neural networks, polynomial kernel SVMs, pretrained Transformer attention-based models, etc. However, even though interactive models are *theoretically capable* of modeling a more expressive set of relationships, it's not clear that they will learn to exploit this additional expressivity, particularly when simpler patterns suffice.

Empirical multimodally-additive projections: EMAP. Given a fixed, trained model f (usually one theoretically capable of modeling visual-textual interactions) and a set of N labelled datapoints $\{(v_i, t_i, y_i)\}_{i=1}^N$, we aim to answer: *does f utilize cross-modal feature interactions to produce more accurate predictions for these datapoints?*

Hooker (2004) gives a general method for computing the projection of a function f onto a set of functions with a specified ANOVA structure (see also Sobol (2001); Liu and Owen (2006)): our algorithmic contributions are to extend the method to multimodal models with $d > 1$ dimensional outputs, and to prove that the multimodal empirical approximation is optimal. Specifically: we are

[2] The methods introduced here can be easily extended beyond just text/image pairs (e.g., to videos, audio, etc.), and to more than two modalites.

[3] The multimedia commonly makes a related distinction between *early fusion* (joint multimodal processing) and *late fusion* (ensembles) (Snoek et al., 2005).

Algorithm 1 Empirical Multimodally-Additive Projection (EMAP); worked example in Appendix G.

```
Input: a trained model $f$ that outputs logits; a set of text/visual pairs $\mathcal{D} = \{(t_i, v_i)\}_{i=1}^{N}$
Output: the predictions of $\hat{f}$, the empirical projection of $f$ onto the set of multimodally-additive functions, on $\mathcal{D}$.

$f_{cache} = 0_{N \times N \times d}$; $\text{preds}_{\hat{f}} = 0_{N \times d}$
for $i, j \in \{1, 2, \ldots, N\} \times \{1, 2, \ldots, N\}$ do
    $f_{cache}(i, j) = f(t_i, v_j)$
end for
$\hat{\mu} \in \mathbb{R}^d = \frac{1}{N^2} \sum_{i,j} f_{cache}(i, j)$
for $i \in \{1, 2, \ldots, N\}$ do
    $\text{proj}_t = \frac{1}{N} \sum_{j=1}^{N} f_{cache}(i, j)$
    $\text{proj}_v = \frac{1}{N} \sum_{j=1}^{N} f_{cache}(j, i)$
    $\text{preds}_{\hat{f}}[i] = \text{proj}_t + \text{proj}_v - \hat{\mu}$
end for
return $\text{preds}_{\hat{f}}$
```

interested in $\tilde{f}$, the following projection of f onto the set of multimodally-additive functions:

$$\tilde{f}(t, v) = \underbrace{\mathop{\mathbb{E}}_{v}[f(t, v)]}_{f_T(t)} + \underbrace{\mathop{\mathbb{E}}_{t}[f(t, v)]}_{f_V(v)} - \underbrace{\mathop{\mathbb{E}}_{t,v}[f(t, v)]}_{\mu}$$

where $\mathbb{E}_v[f(t, v)]$, a function of t, is the *partial dependence* of f on t, i.e., $f_T(t) = \mathbb{E}_v[f(t, v)] = \int f(t, v)p(v)dv$,[4] and similarly for the other expectations. An empirical approximation of the partial dependence function for a given t_i can be computed by looping over all observations:

$$\hat{f}_T(t_i) = \frac{1}{N} \sum_{j=1}^{N} f(t_i, v_j)\,.$$

We similarly arrive at $\hat{f}_V(v_i)$ and $\hat{\mu}$, yielding

$$\hat{f}(t_i, v_i) = \hat{f}_T(t_i) + \hat{f}_V(v_i) + \hat{\mu} \qquad (3)$$

which is what EMAP, Algorithm 1, computes for each (t_i, v_i). Note that Algorithm 1 involves evaluating the original model f on all N^2 $\langle v_i, t_j \rangle$ pairs — even mismatched image/text pairs that do not occur in the observed data. In practice, we recommend only computing this projection over the evaluation set.[5] Once the predictions of f and $\hat{f}$ are computed over the evaluation points, then their performance can be compared according to standard evaluation metrics, e.g., accuracy or AUC.

In Appendix A, we prove that the (mean-centered) sum of empirical partial dependence functions is optimal with respect to squared error, that is:

Claim. *Subject to the constraint that $\hat{f}$ is multimodally additive, Algorithm 1 produces a unique and optimal solution to*

$$\mathop{\arg\min}_{\hat{f}\text{ values}} \sum_{i,j} \|f(t_i, v_j) - \hat{f}(t_i, v_j)\|_2^2. \qquad (4)$$

4 Sanity Check: EMAP Hurts in Tasks that Require Interaction

In §5.1, we will see that EMAP provides a very strong baseline for "unbalanced" multimodal classification tasks. But first, we first seek to verify that EMAP degrades model performance in cases that are designed to require cross-modal interactions.

Synthetic data

We generate a set of "visual"/"textual"/label data (v, t, y) according to the following process:[6]

1. Sample random projection $V \in \mathbb{R}^{d_1 \times d}$ and $T \in \mathbb{R}^{d_2 \times d}$ from $U(-.5, .5)$.
2. Sample $v, t \in \mathbb{R}^d \sim N(0, 1)$; normalize to unit length.
3. If $|v \cdot t| > \delta$ proceed, else, return to the previous step.
4. If $v \cdot t > 0$, then $y = 1$, else $y = 0$.
5. Return the data point (Vv, Tt, y).

This function challenges models to learn whether or not the dot product of two randomly sampled vectors in d dimensions is positive or negative — a task that, by construction, requires modeling the multiplicative interaction of the features in these vectors. To complicate the task, the vectors are randomly projected to two "modalities" of different dimensions, d_1 and d_2 respectively.

We trained a linear model, a polynomial kernel SVM, and a feed-forward neural network on this data: the results are in Table 1. The linear model is additive and thus incapable of learning any meaningful pattern on this data. In contrast, the kernel SVM and feed-forward NN, interactive models, are able to fit the test data almost perfectly. However, when we apply EMAP to the interactive models, as expected, their performance drops to random.

[4] Both Friedman (2001) and Hooker (2004) argue that this expectation should be taken over the marginal $p(v)$ rather than the conditional $p(v|t)$.

[5] When even restricting to the evaluation set would be too expensive, as in the case of the R-POP data we experiment with later, one can repeatedly run EMAP on randomly-drawn 500-instance (say) samples from the test set.

[6] We sample 5K points in an 80/10/10 train/val/test split with $\langle d, d_1, d_2, \delta \rangle = \langle 100, 2000, 1000, .25 \rangle$, though similar results were obtained with different parameter settings.

	Linear (A)	Poly (I)	NN (I)
Test Acc	52.8%	99.6%	99.0%
↳ + EMAP	52.8%	49.4%	53.8%

Table 1: Prediction accuracy on synthetic dataset using additive (A) models, interactive (I) models, and their EMAP projections. Random guessing achieves 50% accuracy. Under EMAP, the interactive models degrade to (close to) random, as desired. See §5 for training details.

	LXMERT	→EMAP	Const Pred
VQA2	70.3	40.5	23.4
GQA	60.3	41.0	18.1

Table 2: As expected, for VQA2 and GQA, the mean accuracy of LXMERT is substantially higher than its empirical multimodally additive projection (EMAP). Shown are averages over $k = 15$ random subsamples of 500 dev-set instances.

Balanced VQA Tasks

Our next sanity check is to verify that EMAP hurts the performance of interactive models on two real multimodal classification tasks that are specifically balanced to require modeling cross-modal feature interactions: VQA 2.0 (Goyal et al., 2017) and GQA (Hudson and Manning, 2019).

First, we fine-tuned LXMERT (Tan and Bansal, 2019), a multimodally-interactive, pretrained, 14-layer Transformer model (See §5 for full description) that achieves SOTA on both datasets. The LXMERT authors frame question-answering as a multi-class image/text-pair classification problem — 3.1K candidate answers for VQA2, and 1.8K for GQA. In Table 2, we compare, in cross-validation, the means of: 1) accuracy of LXMERT, 2) accuracy of the EMAP of LXMERT, and 3) accuracy of simply predicting the most common answer for all questions. As expected, EMAP decreases accuracy on VQA2/GQA by 30/19 absolute accuracy points, respectively: this suggests LXMERT is utilizing feature interactions to produce more accurate predictions on these datasets. On the other hand, performance of LXMERT's EMAP remains substantially better than constant prediction, suggesting that LXMERT's logits do nonetheless leverage some unimodal signals in this data.[7]

5 "Unbalanced" Datasets + Tasks

We now return to our original setting: multimodal classification tasks that have not been specifically formulated to force cross-modal interactions. Our goal is to explore what additional insights EMAP can add on top of standard model comparisons.

We consider a suite of 7 tasks, summarized in Table 3. These tasks span a wide variety of goals, sizes, and structures: some aim to classify semiotic properties of image+text posts, e.g., examining the extent of literal image/text overlap (I-SEM, I-CTX, T-VIS); others are post-hoc annotated according to taxonomies of social interest (I-INT, T-ST1, T-ST2); and one aims to directly predict community response to content (R-POP).[8] In some cases, the original authors emphasize the potentially complex interplay between image and text: Hessel et al. (2017) wonder if "visual and the linguistic interact, sometimes reinforcing and sometimes counteracting each other's individual influence;" Kruk et al. (2019) discuss meaning multiplication, emphasizing that "the text+image integration requires inference that creates a new meaning;" and Vempala and Preoţiuc-Pietro (2019) attribute performance differences between their "naive" additive baseline and their interactive neural model to the notion that "both types of information and their interaction are important to this task."

Our goal is not to downplay the importance of these datasets and tasks; it may well be the case that conditional, compositional interactions occur between images and text, but the models we consider do not yet take full advantage of them (we return to this point in item 6). Our goal, rather, is to provide diagnostic tools that can provide additional clarity on the *remaining* shortcomings of current models and reporting schemes.

Additive and Unimodal Models

The additive model we consider is the linear model from Equation 2 trained over the concatenation of image and text representations.[9] To represent im-

[7] EMAPed LXMERT's performance is comparable to LSTM-based, text-only models, which achieve 44.3/41.1 accuracy on the full VQA2/GQA test set, respectively.

[8] In Appendix B, we include descriptions of our reproduction efforts for each dataset/task, but please see the original papers for fuller descriptions of the data/tasks.

[9] For all linear models in this work, we select optimal hyperparameters according to grid search, optimizing validation model performance for each cross-validation split separately. We optimize: regularization type (L1, L2, vs. L1/L2),

Original paper	Task (structure)	Abbrv.	# image+text pairs we recovered
Kruk et al. (2019)	Instagram		
	↳ intent (7-way classification)	I-INT	1.3K
	↳ semiotic (3-way clf)	I-SEM	1.3K
	↳ contextual (3-way clf)	I-CTX	1.3K
Vempala and Preoţiuc-Pietro (2019)	Twitter visual-ness (4-way clf)	T-VIS	3.9K
Hessel et al. (2017)	Reddit popularity (pairwise ranking)	R-POP	87.2K
Borth et al. (2013)	Twitter sentiment (binary clf)	T-ST1	.6K
Niu et al. (2016)	Twitter sentiment (binary clf)	T-ST2	4.0K

Table 3: The tasks we consider are not specifically balanced to force the learning of cross-modal interactions.

ages, we extract a feature vector from a pretrained EfficientNet B4[10] (Tan and Le, 2019). To represent text, we extract RoBERTa (Liu et al., 2019) token features, and mean pool.[11] Our single-modal baselines are linear models fit over EfficientNet/RoBERTa features directly.

Interactive Models

Kernel SVM. We train an SVM with a polynomial kernel using RoBERTa text features and EfficientNet-B4 image features as input. A polynomial kernel endows the model with capacity to model multiplicative interactions between features.[12]

Neural Network. We train a feed-forward neural network using the RoBERTa/EfficientNet-B4 features as input. Following Chen et al. (2017), we first project text and image features via an affine transform layer to representations t and v, respectively, of the same dimension. Then, we extract new features, feeding the concatenated feature vector $[t; v; v - t; v \odot t]$ to a multi-layer, feed-forward network.[13]

Fine-tuning a Pretrained Transformer. We fine-tuned LXMERT (Tan and Bansal, 2019) for our tasks. LXMERT represents images using 36 predicted bounding boxes, each of which is assigned a feature vector by a Faster-RCNN model (Ren et al., 2015; Anderson et al., 2018). This model uses ResNet-101 (He et al., 2016) as a backbone and is pretrained on Visual Genome (Krishna et al., 2017). Bounding box features are fed through LXMERT's 5-layer Transformer (Vaswani et al., 2017) model. Text is processed by a 9-layer Transformer. Finally, a 5-layer, cross-modal transformer processes the outputs of these unimodal encoders jointly, allowing for feature interactions to be learned. LXMERT's parameters are pre-trained on several cross-modal reasoning tasks, e.g., masked image region/language token prediction, cross-modal matching, and visual question answering. LXMERT achieves high performance on balanced tasks like VQA 2.0: thus, we *know* this model can learn compositional, cross-modal interactions in some settings.[14]

LXMERT + logits: We also trained versions of LXMERT where fixed logits from a pretrained linear model (described above) are fed in to the final classifier layer. In this case, LXMERT is only tasked with learning the residual between the strong additive model and the labels. Our intuition was that this augmentation might enable the model to focus more closely on learning interactive, rather than additive, structure in the fine-tuning process.

5.1 Results

Our main prediction results are summarized in Table 4. For all tasks, the performance of our baseline additive linear model is strong, but we are usually able to find an interactive model that outperforms this linear baseline, e.g., in the case of T-ST2, a polynomial kernel SVM outperforms the linear model by 4 accuracy points. This observation alone *seems* to provide evidence that models

regularization strength (10**(-7,-6,-5,-4,-3,-2,-1,0,1.0)), and loss type (logistic vs. squared hinge). We train models using `lightning` (Blondel and Pedregosa, 2016).

[10] For reproducibility, we used ResNet-18 features for the Kruk et al. (2019) datasets; more detail in Appendix E.

[11] Feature extraction approaches are known to produce competitive results relative to full fine-tuning (Devlin et al., 2019, §5.3); in some cases, mean pooling has been found to be similarly competitive relative to LSTM pooling (Hessel and Lee, 2019).

[12] We again use grid search to optimize: polynomial kernel degree (2 vs. 3), regularization strength (10**(-5,-4,-3,-2,-1,0)), and gamma (1, 10, 100).

[13] Parameters are optimized with the Adam optimizer (Kingma and Ba, 2015). We decay the learning rate when validation loss plateaus. The hyperparameters optimized in grid search are: number of layers (2, 3, 4), initial learning rate (.01, .001, .0001), activation function (relu vs. gelu), hidden dimension (128, 256), and batch norm (use vs. don't).

[14] We follow the original authors' fine-tuning recommendations, but also optimize the learning rate according to validation set performance for each cross-validation split/task separately between 1e-6, 5e-6, 1e-5, 5e-5, and 1e-4.

	I-INT	I-SEM	I-CTX	T-VIS	R-POP	T-ST1	T-ST2
Metric	AUC	AUC	AUC	Weighted F1	ACC	AUC	ACC
Cross-val Setup	5-fold	5-fold	5-fold	10-fold	15-fold	5-fold	5-fold
Constant Pred.	50.0	50.0	50.0	17.2	50.0	50.0	66.2
Prev. SOTA	85.3	69.1	78.8	44	62.7	N/A	70.5
Our image-only	73.6	56.5	61.0	47.2	59.1	73.3	67.2
Our text-only	89.9	71.8	**81.2**	37.6	61.1	69.0	73.1
Neural Network (I)	90.4	69.2	78.5	51.1	63.5	71.1	79.9
Polykernel SVM (I)	**91.3**	**74.4**	**81.5**	50.8	–	72.1	**80.9**
FT LXMERT (I)	83.0	68.5	76.3	**53.0**	63.0	66.4	78.6
↳ + Linear Logits (I)	89.9	73.0	80.7	**53.4**	**64.1**	**75.5**	80.3
Linear Model (A)	90.4	72.8	80.9	51.3	63.7	**75.6**	76.1
Our Best Interactive (I)	**91.3**	**74.4**	**81.5**	**53.4**	***64.2***	**75.5**	**80.9**
↳ + EMAP (A)	**91.1**	**74.2**	**81.3**	51.0	***64.1***	**75.9**	**80.7**

Table 4: Prediction results for 7 multimodal classification tasks. First block: the evaluation metric, setup, constant prediction performance, and previous state-of-the-art results (we outperform these baselines mostly because we use RoBERTa). Second block: the performance of our image only/text only linear models. Third block: the predictive performance of our **(I)**nteractive models. Fourth block: comparison of the performance of the best **(I)**nteractive model to the **(A)**dditive linear baseline. Crucially, we also report the EMAP of the best interactive model, which reveals whether or not the performance gains of the **(I)**nteractive model are due to modeling cross-modal interactions, or not. Italics=computed using 15 fold cross-validation over each cross-validation split (see footnote 5). Bolded values are within half a point of the best model.

are taking advantage of some cross-modal interactions for performance gains. Previous analyses might conclude here, arguing that cross-modal interactions are being utilized by the model meaningfully. *But is this necessarily the case?*

We utilize EMAP as an additional model diagnostic by projecting the predictions of our best-performing interactive models. Surprisingly, for I-INT, I-SEM, I-CTX, T-ST1, T-ST2, and R-POP, EMAP results in *essentially no performance degradation.* Thus, even (current) expressive interactive models are usually unable to leverage cross-modal feature interactions to improve performance. This observation would be obfuscated without the use of additive projections, even though we compared to a strong linear baseline that achieves state-of-the-art performance for each dataset. *This emphasizes the importance of not only comparing to additive/linear baselines, but also to the EMAP of the best performing model.*

In total, for these experiments, we observe a single case, LXMERT + Linear Logits trained on T-VIS, wherein modeling cross-modal feature interactions appears to result in noticeable performance increases — here, the EMAPed model is 2.4 F1 points worse.

How much does EMAP change a model's predictions? For these datasets, a model and its EMAP usually make very similar predictions. For all datasets except T-VIS (where EMAP degrades performance) the best model and its EMAP agree on the most likely label in more than 95% of cases on average. For comparison, retraining the full models with different random seeds results in a 96% agreement on average. (Full results are in Appendix C.)

6 Implication FAQs

Q1: What is your recommended experimental workflow for future work examining multimodal classification tasks?

A: With respect to automated classifiers, we recommend reporting the performance of:

1. A constant and/or random predictor

   to provide perspective on the interplay between the label distribution and the evaluation metrics.

2. As-strong-as-possible single-modal models $f(t, v) = g(t)$ and $f(t, v) = h(v)$

   to understand how well the task can be addressed unimodally with present techniques.

3. An as-strong-as-possible multimodally additive model $f(t, v) = f_T(t) + f_V(v)$

   to understand multimodal model performance without access to sophisticated cross-modal reasoning capacity.

4. An as-strong-as-possible multimodally interactive model, e.g., LXMERT,

   to push predictive performance as far as possible.

5. The EMAP of the strongest multimodally interactive model.

   to determine whether or not the best interactive model is truly using cross-modal interactions to improve predictive performance.

We hope this workflow can be extended with additional, newly developed model diagnostics going forward.

Q2: Should papers proposing new tasks be rejected if image-text interactions aren't shown to be useful?

A: *Not necessarily.* The value of a newly proposed task should not depend solely on how well current models perform on it. Other valid ways to demonstrate dataset/task efficacy: human experiments, careful dataset design inspired by prior work, or real-world use-cases.

Q3: Can EMAP tell us anything fundamental about the type of reasoning required to address different tasks themselves?

A: *Unfortunately, no more than model comparisons can (at least for real datasets).* EMAP is a tool that, like model comparison, provides insights about how specific, fixed models perform on specific, fixed datasets. In FAQ 5, we attempt to bridge this gap in a toy setting where we are able to fully enumerate the sample space.

Q4: Can EMAP tell us anything about individual instances?

A: *Yes; but with caveats.* While a model's behavior on individual cases is difficult to draw conclusions from, EMAP can be used to identify single instances within evaluation datasets for which the model is performing enough non-additive computation to change its ultimate prediction, i.e., for a given (t, v), it's easy to compare $f(t, v)$ to $\mathsf{EMAP}(f(t, v))$: these correspond to the inputs and outputs respectively of Algorithm 1. An example instance-level qualitative evaluation of T-VIS is given in Appendix F.

Q5: Do the strong EMAP results imply that most functions of interest are actually multimodally additive?

Short A: *No.*

Long A: A skeptical reader might argue that, while multimodally-additive models cannot account for cross-modal feature interactions, such feature interactions may not be required for cross-modal reasoning. While we authors are not aware of an agreed-upon definition,[15] we will assume that "cross-modal reasoning tasks" are those that challenge models to compute (potentially arbitrary) logical functions of multimodal inputs. Under this assumption, we show that additive models cannot fit (nor well-approximate) most non-trivial logical functions.

Consider the case of multimodal boolean target functions $f(t, v) \in \{0, 1\}$, and assume our image/text input feature vectors each consist of n binary features, i.e., $t, v = \langle t_1, \dots t_n \rangle, \langle v_1, \dots v_n \rangle$ where $t_i, v_i \in \{0, 1\}$. Our goal will be to measure how well multimodally additive models can fit arbitrary logical functions f. To simplify our analysis in this idealized case: we assume 1) access to a training set consisting of all 2^{2n} input vector pairs, and only measure training accuracy (vs. the harder task of generalization) and 2) "perfect" unimodal representations in the sense that t, v contain all of the information required to compute $f(t, v)$ (this is not the case for, e.g., CNN/RoBERTa features for actual tasks).

For very small cross-modal reasoning tasks, additive models can suffice. At $n = 1$, there are 16 possible binary functions $f(t_1, v_1)$, and 14/16 can be perfectly fit by a function of the form $f_T(t_1) + f_V(v_1)$ (the exceptions being XOR and XNOR). For $n = 2$, non-trivial functions are still often multimodally additively representable; an arguably surprising example is this one: $(t2 \wedge \neg v2) \vee (t1 \wedge t2 \wedge v1) \vee (\neg t1 \wedge \neg v1 \wedge \neg v2)$. But for cross-modal reasoning tasks with more variables,[16] even in this toy setting, multimodally-

[15] Suhr et al. (2019), for example, do not define "visual reasoning with natural language," but do argue that some tasks offer a promising avenue to study "reasoning that requires composition" via visual-textual grounding.

[16] As a lower bound on the number of variables required in a more realistic case, consider a cross-modal reasoning task where questions are posed regarding the counts of, say, 1000 object types in images. It's likely that a separate image variable representing the count of each possible type of object would be required. While an oversimplification, for most cross-modal reasoning tasks, $2n = 6$ variables is still relatively small.

additive models ultimately fall short: for the $n = 3$ case, we sampled over 5 million random logical circuits and none were perfectly fit by the additive models. To better understand how well additive models can *approximate* various logical functions, we fit two types of them for the $n > 2$ case: 1) the EMAP of the input function f directly, and 2) AdaBoost (Freund and Schapire, 1995) with the weak learners restricted to unimodal models.[17] For a reference interactive model, we employ AdaBoost without the additivity restriction.

Figure 2 plots the AUC of the 3 models on the training set; we sample 10K random logical circuits for each problem size. As the number of variables increases, the performance of the additive models quickly decreases (though full AdaBoost gets 100% accuracy in all cases). While these experiments are only for a toy case, they show that EMAPs, and additive models generally, have very limited capacity to compute or approximate logical functions of multimodal variables.

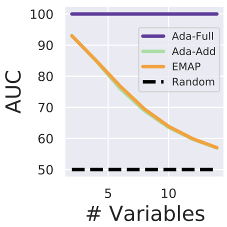

Figure 2: Perf. of additive models on fitting logical circuits.

7 Conclusion and Future Work

The last question on our FAQ list in §6 leaves us with the following conundrum: 1) Additive models are incapable of most cross-modal reasoning; but 2) for most of the unbalanced tasks we consider, EMAP finds an additive approximation that makes nearly identical predictions to the full, interactive model. We postulate the following potential explanations, pointing towards future work:

- Hypothesis 1: These unbalanced tasks don't require complex cross-modal reasoning. This purported conclusion cannot account for gaps between human and machine performance: if an additive model underperforms relative to human judgment, the gap could plausibly be explained by cross-modal feature interactions. But even in cases where an additive model matches or exceeds human performance on a fixed dataset, additive models may still be insufficient. The mere fact that unimodal and additive models *can* often be disarmed by adding valid (but carefully selected) instances post hoc (as in, e.g., Kiela et al. (2020)) suggests that their inductive bias can simultaneously be sufficient for train/test generalization, but also fail to capture the spirit of the task. Future work would be well-suited to explore 1) methods for better understanding which datasets (and individual instances) can be rebalanced and which cannot; and 2) the non-trivial task of estimating additive *human* baselines to compare against.
- Hypothesis 2: Modeling feature interactions can be data-hungry. Jayakumar et al. (2020) show that feed-forward neural networks can require a very high number of training examples to learn feature interactions. So, we may need models with different inductive biases and/or much more training data. Notably, the feature interactions learned even in balanced cases are often not interpretable (Subramanian et al., 2019).
- Hypothesis 3: Paradoxically, unimodal models may be too weak. Without expressive enough single-modal processing methods, opportunities for learning cross-modal interaction patterns may not be present during training. So, improvements in unimodal modeling could feasibly improve feature interaction learning.

Concluding thoughts. Our hope is that future work on multimodal classification tasks report not only the predictive performance of their best model + baselines, but also the EMAP of that model. EMAP (and related algorithms) has practical implications beyond image+text classification: there are straightforward extensions to non-visual/non-textual modalities, to classifiers using more than 2 modalities as input, and to single-modal cases where one wants to check for feature interactions between two groups of features, e.g., premise/hypothesis in NLI.

Acknowledgments

In addition to the anonymous reviewers and meta-reviewer, the authors would like to thank Ajay Divakaran, Giles Hooker, Jon Kleinberg, Julia Kruk, Tianze Shi, Ana Smith, Alane Suhr, and Gregory Yauney for the helpful discussions and feedback. We thank the NVidia Corporation for providing some of the GPUs used in this study. JH performed this work while at Cornell University. Partial support for this work was kindly provided by a grant from The Zillow Corporation, and a Google

[17] AdaBoost is chosen because it has strong theoretical guarantees to fit to training data: see Appendix D.

Focused Research Award. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.

References

Malihe Alikhani, Sreyasi Nag Chowdhury, Gerard de Melo, and Matthew Stone. 2019. CITE: A corpus of image–text discourse relations. In *NAACL*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Roland Barthes. 1988. *Image-music-text*. Macmillan.

John Bateman. 2014. *Text and image: A critical introduction to the visual/verbal divide*. Routledge.

Mathieu Blondel and Fabian Pedregosa. 2016. Lightning: large-scale linear classification, regression and ranking in Python. https://doi.org/10.5281/zenodo.200504.

Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. 2013. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *ACM MM*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *ACL*.

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

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *EMNLP*.

Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*. Springer.

Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 29(5).

Jerome H. Friedman and Bogdan E. Popescu. 2008. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954.

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

Trevor Hastie and Robert Tibshirani. 1987. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Jack Hessel and Lillian Lee. 2019. Something's brewing! Early prediction of controversy-causing posts from discussion features. In *NAACL*.

Jack Hessel, Lillian Lee, and David Mimno. 2017. Cats and captions vs. creators and the clock: Comparing multimodal content to context in predicting relative popularity. In *The Web Conference*.

Giles Hooker. 2004. Discovering additive structure in black box functions. In *KDD*.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. *CVPR*.

Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. 2017. Automatic understanding of image and video advertisements. In *CVPR*.

Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. 2016. Revisiting visual question answering baselines. In *ECCV*. Springer.

Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, and Razvan Pascanu. 2020. Multiplicative interactions and where to find them. In *ICLR*.

James M. Jones, Gary Alan Fine, and Robert G. Brust. 1979. Interaction effects of picture and caption on humor ratings of cartoons. *The Journal of Social Psychology*, 108(2):193–198.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73.

Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in instagram posts. In *EMNLP*.

Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *KDD*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. TVQA: Localized, compositional video question answering. In *EMNLP*.

Jay Lemke. 1998. Multiplying meaning. *Reading science: Critical and functional perspectives on discourses of science*, pages 87–113.

Ruixue Liu and Art B. Owen. 2006. Estimating mean dimensionality of analysis of variance decompositions. *Journal of the American Statistical Association*, 101(474):712–721.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Emily E Marsh and Marilyn Domas White. 2003. A taxonomy of relationships between images and text. *Journal of Documentation*, 59(6):647–672.

Radan Martinec and Andrew Salway. 2005. A system for image–text relations in new (and old) media. *Visual communication*, 4(3):337–371.

Teng Niu, Shiai Zhu, Lei Pang, and Abdulmotaleb El-Saddik. 2016. Sentiment analysis on multi-view social data. In *MultiMedia Modeling*, page 15–27.

Kay O'Halloran. 2004. *Multimodal discourse analysis: Systemic functional perspectives*. A&C Black.

Michael O'Toole. 1994. *The language of displayed art*. Fairleigh Dickinson Univ Press.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. In *Human Interpretability in Machine Learning Workshop at ICML*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*.

Cees G.M. Snoek, Marcel Worring, and Arnold W.M. Smeulders. 2005. Early versus late fusion in semantic video analysis. In *ACM Multimedia*.

Ilya M Sobol. 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280.

Erik Štrumbelj and Igor Kononenko. 2010. An efficient explanation of individual classifications using game theory. *JMLR*.

Sanjay Subramanian, Sameer Singh, and Matt Gardner. 2019. Analyzing compositionality of visual question answering. In *NeurIPS Workshop on Visually Grounded Interaction and Language (ViGIL)*.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *ACL*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

Berk Ustun and Cynthia Rudin. 2016. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Alakananda Vempala and Daniel Preoţiuc-Pietro. 2019. Categorizing and inferring the relationship between the text and image of Twitter posts. In *ACL*.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! Visual object discovery through multi-modal dialogue. In *CVPR*.

Lydia Weiland, Ioana Hulpuş, Simone Paolo Ponzetto, Wolfgang Effelsberg, and Laura Dietz. 2018. Knowledge-rich image gist understanding beyond literal meaning. *Data & Knowledge Engineering*, 117:114–132.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2018. Visual entailment task for visually-grounded language learning. *arXiv preprint arXiv:1811.10582*.

Keren Ye and Adriana Kovashka. 2018. Advise: Symbolism and external knowledge for decoding advertisements. In *ECCV*.

Dani Yogatama and Noah A. Smith. 2015. Bayesian optimization of text representations. In *EMNLP*.

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

Mingda Zhang, Rebecca Hwa, and Adriana Kovashka. 2018. Equal but not the same: Understanding the implicit relationship between persuasive images and text. In *The British Machine Vision Conference (BMVC)*.

A Theoretical Analysis of Algorithm 1

We assume we are given a (usually evaluation) dataset $D = \{\langle t_i, v_i \rangle\}_{i=1}^{n}$ and a trained model f that maps $\langle t_i, v_i \rangle$ pairs to a d-dimensional vector of scores. We seek a multimodally-additive function $\hat{f}$ that matches the values of f on any $\langle t_i, v_j \rangle$ for which there exist v', t' such that $\langle t_i, v' \rangle \in D$ and $\langle t', v_j \rangle \in D$; that is, $\langle t_i, v_j \rangle$ represents any text-image pair we could construct if we decoupled the existing pairs in D.[18]

For simplicity, first assume that $d = 1$ (we handle the $d > 1$ case later). Since $\hat{f}$ is multimodally-additive by assumption, $\exists\, \hat{f}_t, \hat{f}_v$ such that $\hat{f}(t_i, v_j) = \hat{f}_t(t_i) + \hat{f}_v(v_j)$. Our goal is to find the "best" $2n$ values $\hat{f}_t(t_i), \hat{f}_v(v_j)$, or — writing f_{ij} for $f(t_i, v_j)$, τ_i for $\hat{f}_t(t_i)$, and ϕ_j for $\hat{f}_v(v_j)$ for notational convenience — to find τ_i, ϕ_j minimizing

$$\mathcal{L} = \frac{1}{2} \sum_i \sum_j (f_{ij} - \tau_i - \phi_j)^2 \tag{5}$$

Claim 1. *$\mathcal{L}$ is convex.*

Proof. The first-order partial derivatives are:

$$\frac{\partial \mathcal{L}}{\partial \tau_i} = n \cdot \tau_i + \sum_j (\phi_j - f_{ij})$$

$$\frac{\partial \mathcal{L}}{\partial \phi_j} = n \cdot \phi_j + \sum_i (\tau_i - f_{ij})$$

and the Hessian $\mathcal{H}$ is

$$\mathcal{H} = \begin{bmatrix} nI & \mathbf{1} \\ \mathbf{1} & nI \end{bmatrix}, \quad I, \mathbf{1} \in \mathbb{R}^{n \times n}$$

It suffices to show that $\mathcal{H}$ is positive semi-definite, i.e., for any $z \in \mathbb{R}^{2n}$, $z^T \mathcal{H} z \geq 0$. Indeed,

$$\begin{aligned}
z^T \mathcal{H} z &= z^T \begin{bmatrix} nz_1 + \sum_{j=n+1}^{2n} z_j \\ nz_2 + \sum_{j=n+1}^{2n} z_j \\ \vdots \\ nz_n + \sum_{j=n+1}^{2n} z_j \\ nz_{n+1} + \sum_{i=1}^{n} z_i \\ nz_{n+2} + \sum_{i=1}^{n} z_i \\ \vdots \\ nz_{2n} + \sum_{i=1}^{n} z_i \end{bmatrix} \\
&= \sum_{i=1}^{n} \left(nz_i^2 + \sum_{j=n+1}^{2n} z_i z_j \right) \\
&\quad + \sum_{j=n+1}^{2n} \left(nz_j^2 + \sum_{i=1}^{n} z_j z_i \right) \\
&= \sum_{i=1}^{n} \sum_{j=n+1}^{2n} \left(z_i^2 + 2z_i z_j + z_j^2 \right) \\
&= \sum_{i=1}^{n} \sum_{j=n+1}^{2n} (z_i + z_j)^2 \geq 0
\end{aligned}$$

□

Now, for the optimal solutions to our minimization problem, we can set the first-order partial derivatives to 0 and solve for our $2n$ parameters τ_i, ϕ_j. These solutions will correspond to global minima due to the convexity result we established above. It turns out to be equivalent to find solutions to:

$$\mathcal{H} \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \\ \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{n} f_{1k} \\ \vdots \\ \sum_{k=1}^{n} f_{nk} \\ \sum_{k=1}^{n} f_{k1} \\ \vdots \\ \sum_{k=1}^{n} f_{kn} \end{bmatrix} \tag{6}$$

We can do this by finding one solution s to the above, and then analyzing the nullspace of $\mathcal{H}$,

[18] Note that multimodally-additive models do not rely on particular t_i, v_j couplings, as this family of functions decomposes as $f(t_i, v_j) = f_t(t_i) + f_v(v_j)$; thus, a multimodally-additive $\hat{f}$ *should* be able to fit any image-text pair we could construct from D, not just the image-text pairs we observe.

which will turn out to be the 1-dimensional subspace spanned by

$$r = \langle \underbrace{1, 1, \dots 1,}_{n} \underbrace{-1, -1, \dots -1}_{n} \rangle$$

That $\mathcal{H}r = 0$ can be verified by direct calculation. Then, all solutions will have the form $s + cr$ for any $c \in \mathbb{R}$.[19]

Claim 2. *Algorithm 1 computes a solution to Equation 6 as a byproduct.*

Proof. Algorithm 1 computes $s = \langle \tau_i, \phi_j \rangle$ as:

$$\tau_i = \frac{1}{n} \sum_k f_{ik} - \frac{1}{n^2} \sum_i \sum_j f_{ij} \tag{7}$$

$$\phi_j = \frac{1}{n} \sum_k f_{kj} \tag{8}$$

s is a solution to Equation 6, as can be verified by direct substitution. □

Claim 3. *The rank of $\mathcal{H}$ is $2n - 1$, which implies that its nullspace is 1-dimensional.*

Proof. Solutions to $\mathcal{H}x = \lambda x$ occur when:

$$\lambda = n, \; 0 = \sum_{i=1}^{n} x_i \text{ and } 0 = \sum_{j=n+1}^{2n} x_j$$

$$\lambda = 2n, \; x = \mathbf{1}$$

$$\lambda = 0, \; x = r$$

So, zero is an eigenvalue of $\mathcal{H}$ with multiplicity 1, which shows that $\mathcal{H}$'s rank is $2n - 1$.[20] □

Because the nullspace of $\mathcal{H}$ is 1-dimensional, all solutions to Equation 6 are given by $s + cr$ for any $c \in \mathbb{R}$. Returning to the notation of the original problem, we see that all optimal solutions are given by:

$$\tau_i = \frac{1}{n} \sum_k f_{ik} - \frac{1}{n^2} \sum_i \sum_j f_{ij} + c \tag{9}$$

$$\phi_j = \frac{1}{n} \sum_k f_{kj} - c \tag{10}$$

Claim 4. *Algorithm 1 produces a unique solution for the values of $\hat{f}$.*

Proof. We have shown that Algorithm 1 produces *an* optimal solution, and have derived the parametric form of all optimal solutions in Equation 9 and Equation 10. Note that Algorithm 1 outputs $\tau_i + \phi_j$ (rather than τ_i, ϕ_j individually). This cancels out the free choice of c. Thus, any algorithm that outputs optimal $\tau_i + \phi_j$ will have the same output as Algorithm 1. □

Extension to multiple dimensions. So far, we have shown that Algorithm 1 produces an optimal and unique solution for Equation 4, but only in cases where $f_{ij}, \tau_i, \phi_j \in \mathbb{R}$. In general, we need to show the algorithm works for multi-dimensional outputs, too. The full loss function includes a summation over dimension as:

$$\mathcal{L} = \frac{1}{2} \sum_i \sum_j \sum_d (f_{ijd} - \tau_{id} - \phi_{jd})^2 \tag{11}$$

This objective decouples entirely over dimension d, i.e., the loss can be rewritten as:

$$\frac{1}{2} \Bigg(\underbrace{\sum_{ij} (f_{ij1} - \tau_{i1} - \phi_{j1})^2}_{\mathcal{L}_1} + \underbrace{\sum_{ij} (f_{ij2} - \tau_{i2} - \phi_{j2})^2}_{\mathcal{L}_2} + \dots \underbrace{\sum_{ij} (f_{ijd} - \tau_{id} - \phi_{jd})^2}_{\mathcal{L}_d} \Bigg)$$

Furthermore, notice that the parameters in $\mathcal{L}_i$ are disjoint from the parameters in $\mathcal{L}_j$ if $i \neq j$. Thus, to minimize the multidimensional objective in Equation 11, it suffices to minimize the objective for each $\mathcal{L}_i$ independently, and recombine the solutions. This is precisely what Algorithm 1 does.

B Dataset Details and Reproducibility Efforts

B.1 I-INT, I-SEM, I-CTX

This data is available from `https://github.com/karansikka1/documentIntent_emnlp19`. We use the same 5 random splits provided by the authors for evaluation. The

[19] Proof: Assume that s' is a solution of Equation 6. $s' - s$ will be in the nullspace of $\mathcal{H}$. Clearly, $s' = s + (s' - s)$, so s' can be written as $s + x$ for x in the nullspace of $\mathcal{H}$.

[20] We can make explicit the eigenbasis for the $\lambda = n$ solutions. Let $M \in \mathbb{R}^{n \times (n-1)}$ be I_{n-1} with an additional row of -1 concatenated as the n^{th} row. The eigenbasis is given by the columns of

$$\begin{bmatrix} M & \mathbf{0} \\ \mathbf{0} & M \end{bmatrix}.$$

authors provide ResNet18 features, which we use for our non-LXMERT experiments instead of EfficientNet-B4 features. After contacting the authors, they extracted bottom-up-top-down FasterRCNN features for us, so we were able to compare to LXMERT. State of the art performance numbers are derived from the above github repo; these differ slightly from the values reported in the original paper because the github versions are computed without image data augmentation.

B.2 T-VIS

This data is available from https://github.com/danielpreotiuc/text-image-relationship/. The raw images are not available, so we queried the Twitter API for them. The corpus has 4472 tweets in it initially, but we were only able to re-collect 3905 tweets (87%) when we re-queried the API. Tweets can be missing for a variety of reasons, e.g., the tweet being permanently deleted, or the account's owner making the their account private at the time of the API request. A handful of tweets were available, but were missing images when we tried to re-collect them. This can happen when the image is a link to an external page, and the image is deleted from the external page.

B.3 R-POP

This data is available from http://www.cs.cornell.edu/~jhessel/cats/cats.html. We just use the pics subreddit data. We attempted to rescrape the pics images from the imgur urls. We were able to re-collect 87215/88686 of the images (98%). Images can be missing if they have been, e.g., deleted from imgur. We removed any pairs with missing images from the ranking task; we trained on 42864/44343 (97%) of the original pairs. The data is distributed with training/test splits. From the training set for each split, we reserve 3K pairs for validation. The state of the art performance numbers are taken from the original releasing work.

B.4 T-ST1

This data is available from http://www.ee.columbia.edu/ln/dvmm/vso/download/twitter_dataset.html and consists of 603 tweets (470 positive, 133 negative). The authors distribute data with 5 folds pre-specified for cross-validation performance reporting. However, we note that the original paper's best model achieves 72% accuracy in this setting, but a constant prediction baseline achieves higher performance: 470/(470+133) = 78%. Note that the constant prediction baseline likely performs worse according to metrics other than accuracy, but only accuracy is reported. We attempted to contact the authors of this study but did not receive a reply. We also searched for additional baselines for this dataset, but were unable to find additional work that uses this dataset in the same fashion. Thus, given the small size of the dataset, lack of reliable measure of SOTA performance, and label imbalance, we decided to report ROC AUC prediction performance.

B.5 T-ST2

This data is available from https://www.mcrlab.net/research/mvsa-sentiment-analysis-on-multi-view-social-data/. We use the MVSA-Single dataset because human annotators examine both the text and image simultaneously; we chose not to use MVSA-Multiple because human annotators do not see the tweet's image and text at the same time. However, the dataset download link only comes with 4870 labels, instead of the 5129 described in the original paper. We contacted the authors of the original work about the missing data, but did not receive a reply.

We follow the preprocessing steps detailed in Xu and Mao (2017) to derive a training dataset. After preprocessing, we are left with 4041 data points, whereas prior work compares with 4511 points after preprocessing. The preprocessing consists of removing points that are (positive, negative), (negative, positive), or (neutral, neutral), which we believe matches the description of the preprocessing in that work. We contacted the authors for details, but did not receive a reply. The state-of-the-art performance number for this dataset is from Xu et al. (2018).

C Are EMAPs just regularizers?

One reason why EMAPs may often offer strong performance is by acting as a regularizer: projecting to a less expressive hypothesis space may reduce variance/overfitting. But in many cases, the original model and its EMAP achieve similar predictive accuracy. This suggests two explanations: *either* the EMAPed version makes

	I-INT	I-SEM	I-CTX	T-VIS	R-POP	T-ST1	T-ST2
Metric	AUC	AUC	AUC	Weighted F1	ACC	AUC	ACC
Num Classes	7	3	3	4	2	2	2
Setup	5-fold	5-fold	5-fold	10-fold	15-fold	5-fold	5-fold
Best Interactive	Poly	Poly	Poly	LXMERT	LXMERT	LXMERT	Poly
Original Perf.	91.3	74.4	81.5	53.4	*64.2*	75.5	80.9
Original EMAP	91.1	74.2	81.3	51.0	*64.1*	75.9	80.7
DiffSeed Perf.	91.3	74.5	81.4	53.2	*64.1*	75.3	81.3
Match Orig + EMAP	95.6	95.9	97.4	85.5	*96.3*	98.0	96.7
Match Orig + DiffSeed	99.9	99.1	100.0	75.5	*87.6*	92.4	97.9
% Inst. Orig. Better	51.2	52.0	51.5	55.2	*51.2*	5/12 cases ($\approx 6/12 = 50\%$)	53.0

Table 5: Consistency results. The first block provides details about the task and the model that performed best on it. The second block gives the performance (italicized results represent cross-validation EMAP computation results; see footnote 5). The third block gives the percent of time the original model's prediction is the same as for EMAP, and, for comparison, the percent of time the original model's predictions match the identical model trained with a different random seed: in all cases except for T-VIS, the original model and the EMAP make the same prediction in more than 95% of cases. The final row gives the percent of instances (among instances for which the original model and the EMAP disagree) that the original model is correct. Except for T-VIS, when the EMAP and the original model disagree, each is right around half the time.

significantly different predictions with respect the original model (e.g., because it is better regularized), but it happens that those differing predictions "cancel out" in terms of final prediction accuracy; *or*, the original, unprojected functions are quite close to additive, anyway, and the EMAP doesn't change the predictions all that much.

We differentiate between these two hypotheses by measuring the percent of instances for which the EMAP makes a different classification prediction than the full model. Table 5 gives the results: in all cases except T-VIS, the match between EMAP and the original model is above 95%. For reference, we retrained the best performing models with different random seeds, and measured the performance difference under this change.

When EMAP changes the predictions of the original model, does the projection generally change to a more accurate label, or a less accurate one? We isolate the instances where a label swap occurs, and quantify this using: (num orig better) / (num orig better + num proj better). In most cases, the effect of projecting improves and degrades performance roughly equally, at least according to this metric. For T-VIS, however, the original model is better in over 55% of cases: this is also reflected in the corresponding F-scores.

D Logic Experiment Details

In §6, we describe experiments using AdaBoost. We chose AdaBoost (Freund and Schapire, 1995) to fit to the training set because of its strong convergence guarantees. In short: if AdaBoost can find a weak learner at each iteration (that is: if it can find a candidate classifier with above-random performance) it will be able to fit the training set. A more formal statement of AdaBoost's properties can be found in the original work.

The AdaBoost classifiers we consider use decision trees with max depth of 15 as base estimators. We choose a relatively large depth because we are not concerned with overfitting: we are just measuring training fit. The additive version of AdaBoost we consider is identical to the full AdaBoost classifier, except, at each iteration, either the image features or the text features individually are considered.

E Additional Reproducibility Info

Computing Hardware. Linear models and feed-forward neural networks were trained using consumer-level CPU/RAM configurations. LXMERT was fine-tuned on single, consumer GPUs with 12GB of vRAM.

Runtime. The slowest algorithm we used was LXMERT (Tan and Bansal, 2019), and the biggest

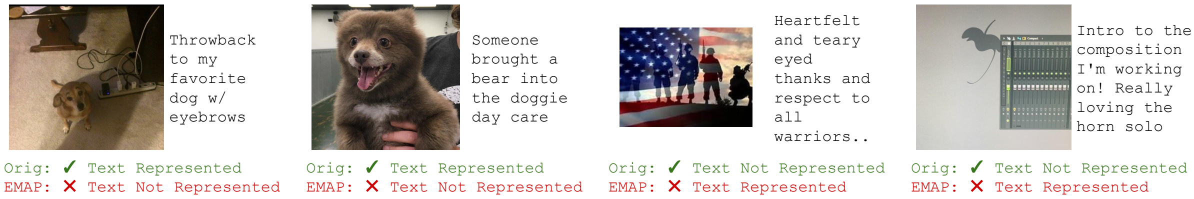
Throwback to my favorite dog w/ eyebrows
Orig: ✓ Text Represented
EMAP: ✗ Text Not Represented
Someone brought a bear into the doggie day care
Orig: ✓ Text Represented
EMAP: ✗ Text Not Represented
Heartfelt and teary eyed thanks and respect to all warriors..
Orig: ✓ Text Not Represented
EMAP: ✗ Text Represented
Intro to the composition I'm working on! Really loving the horn solo
Orig: ✓ Text Not Represented
EMAP: ✗ Text Represented

Figure 3: Examples of cases from for which EMAP degrades the performance of LXMERT + Logits. All cases are labelled as "image does not add meaning" in the original corpus. Text of tweets may be gently modified for privacy reasons.

dataset we fine-tuned on was R-POP. LXMERT was fine-tuned on the order of 1500 times. Depending on the dataset, the fine-tuning process took between 10 minutes and an hour. Overall, we estimate that we spent on the order of 50-100 GPU days doing this work.

Number of Parameters. The RoBERTa features we used were 2048-dimensional, and the EfficientNetB4 features were 1792 dimensional. The linear models and feed forward neural networks operated on those. We cross-validated the number of layers and the width, but the maximal model, for these cases, has on the order of millions of parameters. The biggest model we used was LXMERT, which has a comparable memory footprint to the original BERT Base model.

F Qualitative Analysis of T-VIS

To demonstrate the potential utility of EMAP in qualitative examinations, we identified the individual instances in T-VIS for which EMAP changes the test-time predictions of the LXMERT + Linear Logits model. Recall that in this dataset, EMAP hurts performance.

In introducing this task, Vempala and Preoţiuc-Pietro (2019) propose categorizing image+text tweets into four categories: "Some or all of the content words in the text are represented in the image" (or not) × "Image has additional content that represents the meaning of the text and the image" (or not).

As highlighted in Table 5, when EMAP changes the prediction of the full model (14.5% of cases), the prediction made is incorrect more often not: among label swapping cases, when the EMAP or the original model is correct, the original prediction is correct in 55% of the cases.

The most common label swaps of this form are between the classes: "image does not add" × {"text is represented", "text is not represented"}; as shorthand for these two classes, we will write IDTR ("image doesn't add, text represented") and IDTN ("image doesn't add, text not represented"). Across the 10 cross-validation splits, EMAP incorrectly maps the original model's correct prediction of INTR → IDTN 255 times. For reference, there are 165 cases where EMAP maps the *incorrect* INTR prediction of the original model to the correct IDTN label. So, when EMAP makes the change INTR → IDTN, in 60% of cases the full model is correct. Similarly, EMAP incorrectly maps the original model's correct prediction of INTN → IDTR 77 times (versus 48 correct mappings, original model correct in 62% of cases).

Figure 3 gives some instances from T-VIS where images do not add meaning and the EMAP projection causes a swap from a correct to an incorrect prediction. While it's difficult to draw conclusions from single instances, some preliminary patterns emerge. There are a cluster of animal images coupled with captions that may be somewhat difficult to map. In the case of the dog with eyebrows, the image isn't centered on the animal, and it might be difficult to identify the eyebrows without the prompt of the caption (hence, interactions might be needed). Similarly, the bear-looking-dog case is difficult: the caption doesn't explicitly mention a dog, and the image itself depicts an animal that isn't canonically canine-esque; thus, modeling interactions between the image and the caption might be required to fully disambiguate the meaning.

Figure 3 also depicts two cases where the original model predicts that the text is not represented (but EMAP does). They are drawn from a cluster of similar cases where the captions seem indirectly connected to the image. Consider the 4th, music composition example: just looking at the text, one could envision a more literal manifestation: i.e., a person playing a horn. Similarly, looking just at the screenshot of the music production software, one could envision a more literal caption, e.g., “producing a new song with this software.” But, the real connection is less direct, and might require additional cross-modal inferences. Other common cases of INTR → IDTN are “happy birthday” messages coupled with images of their intended recipients and selfies taken at events (e.g., sports games), with descriptions (but not direct visual depictions).

G Worked Example of EMAP

We adopt the notation of Appendix A and give an concrete worked example of EMAP. Consider the following f output values, which are computed on three image+text pairs for a binary classification task. We assume that f outputs an un-normalized logit that can be passed to a scaling function like the logistic function σ for a probability estimate over the binary outcome, e.g., $f_{11} = -1.3$ and $\sigma(f_{11}) = P(y = 1) \approx .21$.

$$\begin{array}{ccc} f_{11} = -1.3 & f_{12} = .3 & f_{13} = -.2 \\ f_{21} = .8 & f_{22} = 3 & f_{23} = 1.1 \\ f_{31} = 1.1 & f_{32} = -.1 & f_{33} = .7 \end{array}$$

We can write this equivalently in matrix form:

$$\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = \begin{bmatrix} -1.3 & 0.3 & -0.2 \\ 0.8 & 3.0 & 1.1 \\ 1.1 & -.1 & 0.7 \end{bmatrix}$$

Note that the mean logit predicted by f over these 3 examples is .6. We use Equation 8 to compute the ϕ_js; this is equivalent to taking a column-wise mean of this matrix, which yields (approximately) $[.2, 1.07, .53]$. Similarly, we can use Equation 7, equivalent to taking a row-wise mean of this matrix, which yields (approximately) $[-.4, 1.63, .57]$, and then subtract the overall mean .6 to achieve $[-1, 1.03, -.03]$. Finally, we can sum these two results to compute $[\hat{f}_{11}, \hat{f}_{22}, \hat{f}_{33}] = [-.8, 2.1, .5]$. These predictions are the closest approximations to the full evaluations $[f_{11}, f_{22}, f_{33}] = [-1.3, 3, .7]$ for which the generation function obeys the additivity constraint over the three input pairs.

References

Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*. Springer.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Alakananda Vempala and Daniel Preoţiuc-Pietro. 2019. Categorizing and inferring the relationship between the text and image of Twitter posts. In *ACL*.

Nan Xu and Wenji Mao. 2017. MultiSentiNet: A deep semantic network for multimodal sentiment analysis. In *CIKM*.

Nan Xu, Wenji Mao, and Guandan Chen. 2018. A co-memory network for multimodal sentiment analysis. In *SIGIR*.