Supplementary material

A Application to VQA

Data. We generate the counterfactual examples by masking image features on-the-fly, during training, according the the human attention maps of [18]. We use image features from [3], which correspond to bounding boxes in the image. We mask the features whose boxes overlap with a fraction of the the human attention map above a fixed threshold. We use the precomputed overlap score from [57], which is a scalar in $[0, 1]$, and set the threshold at 0.2 (setting it at 0 would mask the occasional boxes that encompass nearly the whole image, which is not desirable). This value was set manually by verifying for the intended effect on a few training examples (that is, masking most of the relevant visual evidence). See Fig. 6 for examples of original questions and their counterfactual versions.

Experimental setting. Our experiments use a validation set (8,000 questions chosen at random) held out from the original VQA-CP training set. Note that most existing methods evaluated in VQA-CP use the extremely unsanitary practice of using the VQA-CP test split for model selection. This is extremely concerning since the whole purpose of VQA-CP is to evaluate generalization to an out-of-distribution test set. The variance in evaluating the 'number' and 'yes/no' questions is moreover extremely high, because the number of reasonable answers on each of these types is very limited. For example, a model that answers *yes* or *no* at random, or produces constantly either answer, can fare extremely well (upwards of 62% accuracy) on these questions. This can very well result from a buggy implementation or a "lucky" random seed, identified by model selection on the test set (!). This is the reason why we include an evaluation on the 'other' type of questions in isolation. All of these issues have been pointed out by a few authors [26, 16, 66].

Our *focused* test set is a subset of the official VQA-CP test set. It is created in a similar manner as the counterfactual examples. We mask features that overlap with human attention maps *below* (instead of above) a threshold of 0.8. This value was set manually by verifying for the intended effect on a few examples (masking the background but not the regions necessary to answer the question). The *focused* test set is much smaller than the official test set since it only comprises questions for which a human attention map is available.

Models. Our baseline model follows the general description of Teney *et al.* [64]. We use the features of size 36×2048 provided by Anderson *et al.* [3]. Our 'strong baseline' uses the additional procedure described in [17] on top of this baseline, using the code provided by the authors.

Existing methods. The method presented in [57] could have constituted an ideal point of comparison with ours, as it was evaluated on VQA-CP and used human attention maps. However, after extensive discussions with the authors,

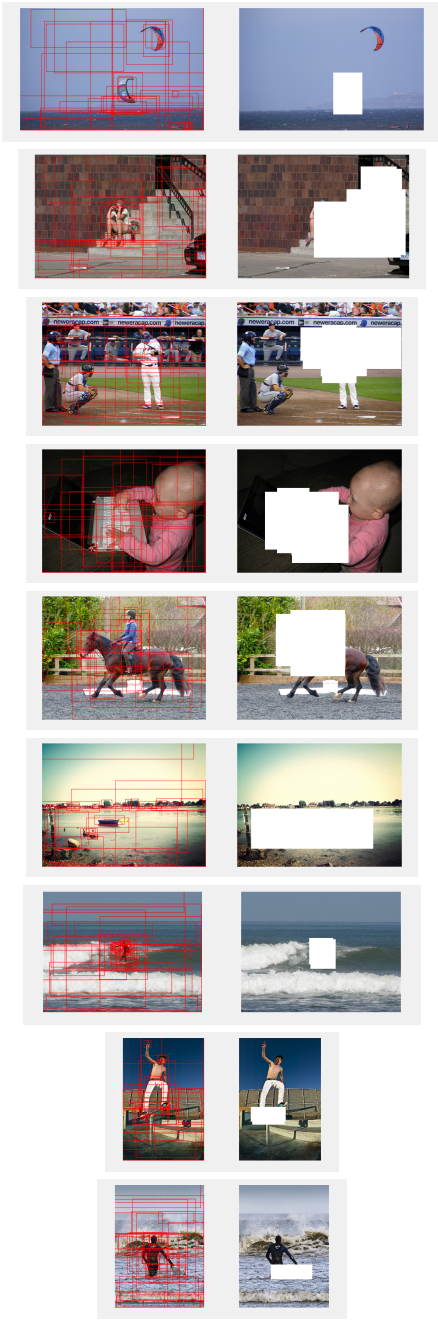


Fig. 6. Application to VQA. Examples of original examples (with their ground truth answer) and their counterfactual version. Red boxes indicate regions that were candidates for masking when generating the counterfactual versions.

we still have not been able to replicate any of the performance claimed in the paper. We found a number of errors in the paper, as well as inconsistencies in the reported results, and an extreme sensitivity to a single hyperparameter (their reported results were obtained with a single run on a single random seed). We chose not to mention this work in our main paper until these issues have been resolved.

Why not use the same technique for the VQA and COCO experiments ? Inpainting in pixel space *vs* masking image features. The two approaches are applicable in both cases. The only reason was to showcase the use of multiple techniques to generate counterfactual examples. The human attention map are specific to VQA and not applicable to the COCO experiments.

B Application to image classification with COCO

Data. We use the edited images released by [1] together with the corresponding original images from COCO. The edited images were created with the inpainter GAN [59] to mask ground truth bounding boxes of specific objects. The images come from the COCO splits *train2014* and *val2014*. We keep this separation for our experiments as follows. Images from *train2014* (323,116 counting original and edited ones) are used for training, except a random subset (1,000 images) that we hold out for validation (model selection, early stopping). Images from *val2014* (3,361 original and 3,361 edited) are used exclusively for testing.

We identified a subset (named *Hard edited*) of the edited images from *val2014* whose ground truth vector (which indicated the classes appearing in the image) is never seen during training (614 images).

The set of edited images provided by [1] is a non-standard subset of COCO, so no directly-comparable results have been published for the multi-label classification task that we consider.

Model. We pre-extract image features from all images with the ResNet-based, bottom-up attention model [3]. These features are averaged across spatial locations, giving a single vector of dimensions 2048 to represent each image. Our model is a 3-layer ReLU MLP of size 64, followed by a linear/sigmoid output layer of size 80 (corresponding to the 80 COCO classes). This baseline model was first tuned for best performance on the validation set (tuning the number of a layers and their size, the batch size, and learning rate), before adding the proposed GS loss. The model is optimized with AdaDelta, mini-batches of size 512, and a binary cross-entropy loss.

Performance is measured with a standard mean average precision (mAP) (as defined in the Pascal VOC challenge) over all 80 classes.

The Fig. 4 in the paper shows the input image with the scores of the top-k predicted labels by the baseline and by our method. The k corresponds to the number of ground truth labels of each image.

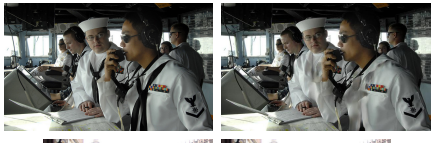
Masked object: **car**
(left and right, behind the truck)

Masked object: **person**

Masked object: **skateboard**

Masked object: **surfboard**

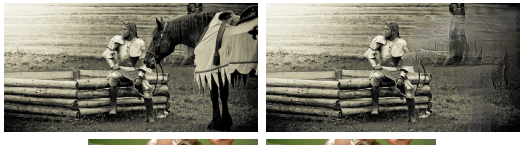
Masked object: **boat**



Masked object: **tie**
(on both persons in the foreground)

Masked object: **bicycle**
(against the railing on the right)

Masked object: **person**



Masked object: **horse**

Masked object: **tie**

Fig. 7. Application to multi-label image classification with COCO. Examples of original and edited images.

Random baseline. In our ablations, this model is identical to the standard baseline, but it is trained with a randomly shuffled training set. We shuffle the inputs $\{\boldsymbol{x}_i\}_i$ and the ground truth labels $\{y_i\}_i$ of all training examples. The model is thus not getting any relevant training signal from any example. It can only leverage static dataset biases (*i.e.* a class imbalance).

C Application to NLP tasks

Sentiment analysis data. We use the subset of the IMDb dataset [39] for which Kaushik *et al.* [34] obtained counterfactual examples. We use their 'paired' version of the data, which only contains original examples that do have an edited version. For **training**, we use the 'train' split of original and edited data (3414 examples). For **validation** (model selection, early stopping), we use the 'dev' set of paired examples. For **testing**, we use the 'test' split, reporting accuracy over the original and edited examples separately. For testing on other datasets, we use a random subset (2000 examples) of the test sets of Amazon Reviews [45], Semeval 2017 (Twitter data) [55], and Yelp reviews [77] similarly to [34].

Sentiment analysis model. We first optimized a simple baseline model on the validation set (tuning the number of a layers, embedding sizes, batch size, and learning rate). We then added the proposed gradient supervision, tuned its hyperparameters on the validation set (regularizer weight) then reported the performance on the test sets at the epoch of best performance on the validation set. The sentences are tokenized and trimmed to a maximum of 32 tokens. The model encodes a sentence as a bag of words, using word embeddings of size 50, averaged to the exact length of each sentence (*i.e.* not including the padding of the shorter sentences). The vocabulary is limited to the 20,000 most frequent words in the dataset. The averaged vector is passed to a simple linear classifier with a sigmoid output. All weights, including word embeddings, are initialized from random values, and optimized with AdaDelta, in mini-batches of size 32, with a binary cross-entropy loss. The best weight for the GS regularizer was found to be λ=20. To reduce the noise in the evaluation due to the small size of the training set, we use an ensemble of 6 identical models trained in parallel. The reported results uses the output of the ensemble, that is the average of the logits of the 6 models.

NLI data. The experiments on NLI follow a similar procedure to those on sentiment analysis. We use the subset of the SNLI dataset [10] for which Kaushik *et al.* [34] collected counterfactual examples. We use their biggest version of the data, named 'all combined', that contains counterfactual examples with edited premises and edited hypotheses. For testing, we evaluate accuracy separately on original and edited examples (edited premises and edited hypotheses combined). For testing transfer, we use the 'dev' set of MultiNLI [73]. Whereas the SNLI dataset contains sentence pairs derived from image captions, MultiNLI is more diverse. It contains sentences from transcribed speech, popular fiction, and government reports. Compared to SNLI, it contains more linguistic diversity and complexity.

Test data →	Yelp
Random predictions (chance)	45.4
Baseline w/o edited tr. data	82.8
Baseline w/ edited tr. data	87.4
+ **GS**, counterfactual rel.	**88.8**
+ GS, random relations	57.4

Table 4. Application to sentiment analysis. Results on the Yelp dataset. This column was missing in Table 3 in the paper (a code-generating error replicated the values from the *Amazon* column into the *Yelp* column).

NLI model. The premise and hypothesis sentences are tokenized and trimmed to a maximum of 32 tokens. They are encoded separately as bags of words, using frozen Glove embeddings (dimension 300), then a learned linear/ReLU projection to dimension 50, and an average to the length of each sentence (without using the padding). They are then passed through a batch normalization layer, then concatenated, giving a vector of size 100. The vector is passed through 3 linear/ReLU layers, then a final linear/sigmoid output layer. The model is trained with AdaDelta, with mini-batches of size 512, and a binary cross-entropy loss. The best weight for the GS regularizer was found to be λ=0.01. Similarly to our experiments on sentiment analysis, we evaluate an ensemble of 6 copies of the model described above.