Supplementary:
Barlow constrained optimization for Visual Question Answering

Abhishek Jha[1*] Badri Patro[1*] Luc Van Gool[1,2] Tinne Tuytelaars[1]
[1]ESAT-PSI, KU Leuven, [2]CVL, ETH Zürich
firstname.lastname@esat.kuleuven.be

Code: https://github.com/abskjha/Barlow-constrained-VQA

1. Overview

In this supplementary, we provide additional details on our proposed models and experiments. In Section 2, we discuss the baseline GGE model [4] and compare its architecte with our proposed Constrained Optimization with Barlow (COB) model. In Section 3 we provide the details on the implementation, datasets used for training and evaluation, architectural details, and the hyperparameters. In Section 3.1, we discuss the algorithms for our models. Section 3.2 has a detailed analysis of hyperparameters (λ, κ, $step$ size). We also extended Section 5.3 from the main paper by providing more details on the selection of Barlow projector's output dimensionality, N_B. Additional qualitative and explainability results are provided in Section 4. Finally, we provide a list of all the mathematical notations used in the main paper and supplementary in the glossary, Section 4.1.

2. Brief discussion on GGE-DQ-iter Model.

We use GGE-DQ-iter[4] as our baseline model. This model consists of an image encoder and text encoder for image and question input, respectively, similar to the UpDn[2] architecture. The GGE model uses a self-attention network to get the joint feature representation by combining image encoded and question encoded features. Finally, a classifier network to predict the answer for the given image and question input. The GGE-DQ-iter method uses a two-stage training mechanism to train the model. In the first stage, the model tries to overcome the question bias, and the second stage it tries to overcome the distributional bias in an iterative fashion. Figure 1 shows the block diagrams for the baseline GGE-DQ model and our COB model built upon this base GGE-DQ architecture. A detailed analysis of the loss function and exact model details are available in the Han *et al*. [4].

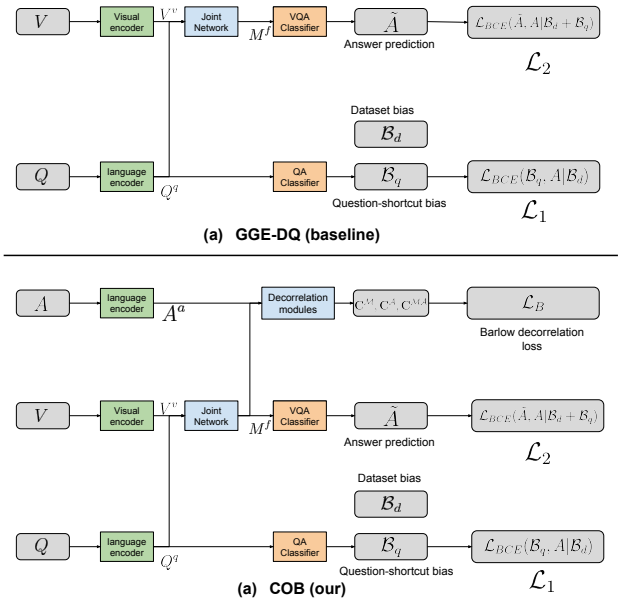

Figure 1: **GGE-DQ v/s COB:** (a) shows the baseline GGE-DQ model [4], (b) shows our proposed COB model built upon the base GGE-DQ model. In the main paper, we formulate COB and ATB over the cross-entropy loss $\mathcal{L}_{CE}$ for a generic classification-based VQA model. However, GGE-DQ uses binary cross-entropy(BCE) as the categorical loss. It models two biases in its loss: a distribution bias $\mathcal{B}_d$ and a question-shortcut bias $\mathcal{B}_d$. Conditioned upon these biases, two BCE losses are computed $\mathcal{L}_1, \mathcal{L}_2$ for the question-only stream and the vision+question stream, respectively. Hence, to build our COB with GGE-DQ as the base architecture, we also use BCE loss, as shown in (b). The constraint formulation and balancing of losses remain the same, as proposed in the main paper, for the generic VQA model. A detailed discussion on the dataset bias and question-shortcut bias can be found in Han *et al*. [4].

*Equal contribution.

3. Implementation details

Dataset To evaluate our proposed model, we conduct experiments on the standard VQA v2 [3] and language-bias sensitive VQA-CP v2 [1] datasets. VQA v2 dataset contains 443K train, 214K val, and 453K test question-answer pairs corresponding to 83K train, 40K val, and 81K test images sampled from MS COCO datasets. VQA-CP v2 contains the same data as VQA v2 while overcoming its language bias by restructuring the answers and questions in the training and the validation sets, such that prior distribution of answers for every question type in the train and validation set differ from each other. The redistribution of data makes VQA-CP v2 more balanced and robust to language bias.

Architecture: In our model, we use Bottom-Up and Top-Down (UpDn)[2] features as input for image representation, and GloVe [7] based word embedding for question tokens input followed by an LSTM [5] to obtain Question representation. We use an attention mechanism to combine visual feature and question representation to obtain joint representation, followed by a classifier to obtain answer logits. For each example (consisting of image, question, and answer) in the VQA dataset, we obtain a joint embedding of image-and-question and an answer token embedding based on the GloVe word embedding model. We use a two-stream model between the joint representation and the answer representation. We project the joint representation and the answer representation to a latent embedding space using a projector network, as shown in Figure 2 of the main paper. The projector network has two linear layers, each of dimension 512 output units. The first layer of the network consists of a linear layer followed by a rectified linear unit followed by a second linear layer. We add the output of the first linear and second layers, followed by a normalization layer to get the final projection embedding. The joint representation is used for the answer prediction task, and the projected embeddings are fed to the Barlow decorrelation loss function.

Decorrelation in Barlow space: In Figure 2, we visualize the correlation matrices in the N_B dimensional Barlow space where decorrelation loss ($\mathcal{L}_B$) is computed. We observe that, for a randomly initialized network, the correlation matrix shows a higher redundancy, as shown by the similar values of the diagonal elements as that of the non-diagonal elements, i.e., a non-prominent diagonal for ($C^{\mathcal{M}}, C^{\mathcal{A}} and C^{\mathcal{MA}}$). At convergence, both ATB and COB models (middle and bottom rows of Figure 2) show (i) a prominent diagonal in auto-correlation matrices ($C^{\mathcal{M}}, C^{\mathcal{A}}$), which means the feature components share less information with other feature components and thus being more informative. (ii) A prominent diagonal in cross-correlation matrix ($C^{\mathcal{MA}}$), implies that our multimodal Barlow decorrelation loss aligns the two modalities (join-embedding and answers) along the major components while keeping the individual feature components decorrelated with each other.

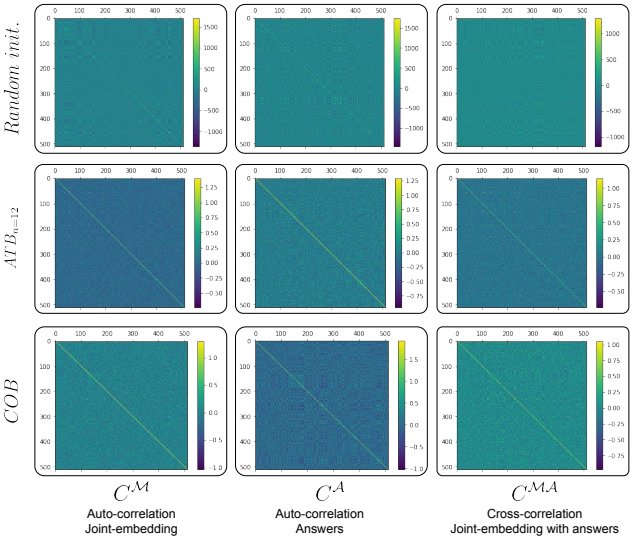

Figure 2: **Decorrelation in Barlow space:** Figure shows the auto-correlation and cross-correlation matrices for a randomly initialized network (top-row), $ATB_{n=12}$ model (middle-row) at convergence and COB model (bottom-row) at convergence. Barlow decorrelation loss forces the feature components to share less information with other feature components by decorrelating them, as can be seen by higher value diagonal elements in the auto-correlation matrices ($C_{\mathcal{M}}$ and $C_{\mathcal{A}}$) at convergence for both ATB and COB models. Our proposed multimodal Barlow decorrelation loss ($\mathcal{L}_B^{\mathcal{MA}}$) also forces the two modalities to be aligned along their major component axes while being decorrelated along the feature dimension, as can be seen by a prominent diagonal in the cross-correlation matrix ($C_{\mathcal{MA}}$).

This alignment of features between the two modalities helps the underlying joint-embedding to learn the semantics of the answer space (embedded in the GloVe word embedding space), which is otherwise not possible by using only the categorical loss.

Analysis on the amount of pre-training for ATB: Here, we extend Section 4.2 of the main paper. In Table 1, we present additional VQA results using our ATB model for different pre-training epochs, n. We evaluate on different types of questions sets from the VQA-CP v2 [1] dataset, namely: Y/N, Number, and Other, along with the overall results on all of these sets.

3.1. Algorithms

To elaborate the formulation and training policies for the proposed ATB and COB models, we provide the respective algorithms in *Algorithm* 1 and 2. All the mathematical notations are defined in Section 3 and visually placed in Figure 2. We also provide a glossary of all the notations in Section 4.1.

Table 1: Ablation analysis: Applying Barlow loss after certain epoch. All the results are % answering accuracy on VQA-CP v2 test set.

Method	All	Y/N	Number	Other
baseline (GGE)	**56.08**	86.64	22.15	49.38
$ATB_{n=0}$	53.64	87.58	14.94	46.47
$ATB_{n=2}$	55.19	85.51	20.22	48.89
$ATB_{n=4}$	55.60	86.29	23.94	48.21
$ATB_{n=6}$	56.75	87.57	22.82	49.90
$ATB_{n=8}$	56.76	87.81	23.08	49.73
$ATB_{n=10}$	56.74	87.70	23.14	49.73
$ATB_{n=11}$	57.16	87.34	**27.45**	**49.53**
$ATB_{n=12}$	**57.18**	**87.53**	27.19	49.51
$ATB_{n=13}$	56.80	87.71	24.34	49.50
$ATB_{n=14}$	56.77	87.62	23.85	49.53
$ATB_{n=16}$	56.59	87.10	23.90	49.58
$ATB_{n=18}$	56.38	87.94	20.98	49.57

ALGORITHM 1. Align then Barlow (ATB)

Input : Batches (V,Q,A), n
Parameters: $\theta_J, \theta_L,$ and $\theta_B = \{\theta_{B_M}, \theta_{B_A}\}$
Result : Learned parameters $\theta_J, \theta_L, \theta_B$.
Initialize $epoch = 0$;
while *is training* **do**
  Compute categorical loss for the current batch, $\mathcal{L}_{CE}$;
  if $epoch \leq n$ **then**
    Compute gradients $G_{\theta_J} \leftarrow \frac{\partial \mathcal{L}_{CE}}{\partial \theta_J}$ and $G_{\theta_L} \leftarrow \frac{\partial \mathcal{L}_{CE}}{\partial \theta_L}$;
    Update parameters as $\triangle_{\theta_J,\theta_L} \propto -G_{\theta_J,\theta_L}$;
  else
    Compute Barlow decorrelation loss for the current batch, $\mathcal{L}_B$;
    $\mathcal{L}_{ATB} \leftarrow \frac{1}{2}(\mathcal{L}_{CE} + \mathcal{L}_B)$;
    Compute gradients $G_{\theta_J} \leftarrow \frac{\partial \mathcal{L}_{ATB}}{\partial \theta_J}$, $G_{\theta_L} \leftarrow \frac{\partial \mathcal{L}_{ATB}}{\partial \theta_L}$ and $G_{\theta_B} \leftarrow \frac{\partial \mathcal{L}_{ATB}}{\partial \theta_B}$;
    Update parameters as: $\triangle_{\theta_J,\theta_L,\theta_B} \propto -G_{\theta_J,\theta_L,\theta_B}$;
  end
  $epoch \leftarrow epoch + 1$;
end

ALGORITHM 2. Constrained optimization with Barlow (COB)

Input : Batches (V,Q,A), κ, $step$
Parameters: $\theta = \{\theta_J, \theta_L, \theta_{B_M}, \theta_{B_A}\}$, Lagrange multiplier λ
Result : Learned parameters θ, and Lagrange multipliers λ.
Initialize $t = 0, \lambda = 1e-5$;
while *is training* **do**
  Compute categorical loss for the current batch, $\mathcal{L}_{CE}$;
  Compute the constraint loss over the batch, $\mathbb{C}^t \leftarrow (\mathrm{L}_B - \kappa)$;
  Compute gradient $G_\theta \leftarrow \frac{\partial \mathcal{L}_{all_{COB^\lambda}}}{\partial \theta}$
  Update parameters as $\triangle_\theta \propto -G_\theta$;
  Update $\mathbb{C}^t$, as: `// Required to compute` λ_{t+1}
  if *t == 0* **then**
    $\mathbb{C}^t \leftarrow (\mathrm{L}_B - \kappa)$;
  else
    $\mathbb{C}^t \leftarrow \alpha\mathbb{C}^{t-1} + (1-\alpha)(\mathrm{L}_B - \kappa)$;
  end
  Update λ, as:
  if $t\%step == 0$ **then**
    $\lambda_{t+1} \leftarrow \lambda_t \exp(StopGradient(\mathbb{C}^t))$;
    `// Update` λ
  else
    $\lambda_{t+1} \leftarrow \lambda_t$;
  end
  $t \leftarrow t + 1$;
end

3.2. Hyperparameter Selection

Selection of λ : We perform our experiment for different values of λ. We observe that for the large value of λ the loss function does not converge, and for a small value of λ, the loss function converges at its optimum performance. We start λ value from 0.1 and increased in its order of magnitude up to $\lambda = 1e-8$ value. We observe that for $\lambda = 1e-5$, the model performs best out of all other values of λ, as shown in the Figure 3.

Selection of *step* size : The tuning of the hyperparameter, step size, (Number of Iteration) plays a crucial role in training our COB model. The λ value updates after a specific step size. We perform our experiment with different values of step sizes (Number of Iteration) starting from step size 50 to step size 800 as shown in Figure 4. We observe that step size 100 performs better than other step sizes.

Selection of κ : κ is the threshold value which controls when the lambda value starts decreasing. When the κ value reaches the Barlow twin loss value, the constraint value becomes zero, and after that, the constraint becomes negative. The negative constraint value tries to reduce the contribution of Barlow twin loss in the total loss value. The selection of κ value is a complex task. We need to observe the pre-trained model and set the κ value to the saturation value of the Barlow loss(where Barlow loss does not change much). Set the κ value near to that saturation value. We experimented with analyzing the behaviors of κ we set with higher saturation value and lower saturation value as shown

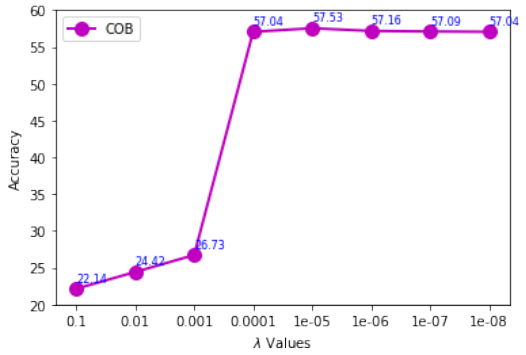

Figure 3: Tuning for hyperparameter Lagrange multiplier λ.

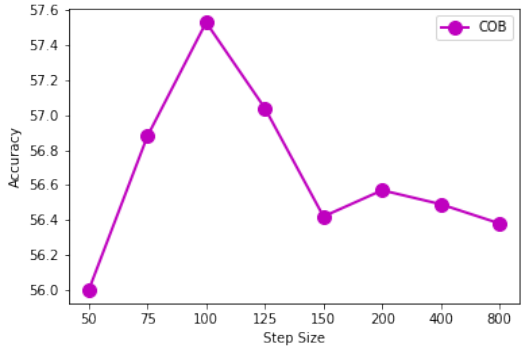

Figure 4: Tuning for hyperparameter Step size.

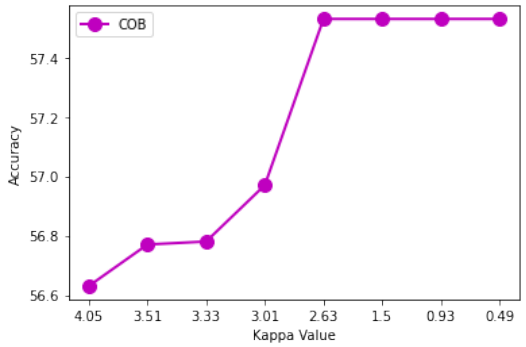

Figure 5: Tuning for hyperparameter κ.

in Figure 5. Based on the empirical observation we select $\kappa = 2.63$ for training COB model. We observe, for lower saturation value, the performance does not affect much.

Selection of N_B : N_B is the output dimensionality of the Barlow projectors ($b_{\theta_{B_M}}, b_{\theta_{B_A}}$). It is an important hyperparameter, as a too-small value to N_B leads to a smaller Barlow space where the multiple semantic concepts would be required to be modelled by the same feature component, and a too-large value would cause multiple feature components to model the same semantic concept. These two cases result in inferior performances, as shown in Figure 6. Here,

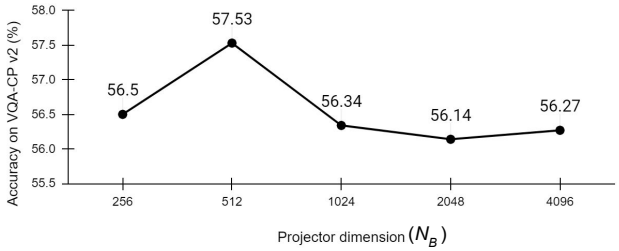

Figure 6: **Effect of the dimensionality of the projector network on the answering performance:** Answering performance on VQA-CP v2 dataset using COB model with different projector dimensions. Best performing model has a projector with output dimensionality $N_B = 512$.

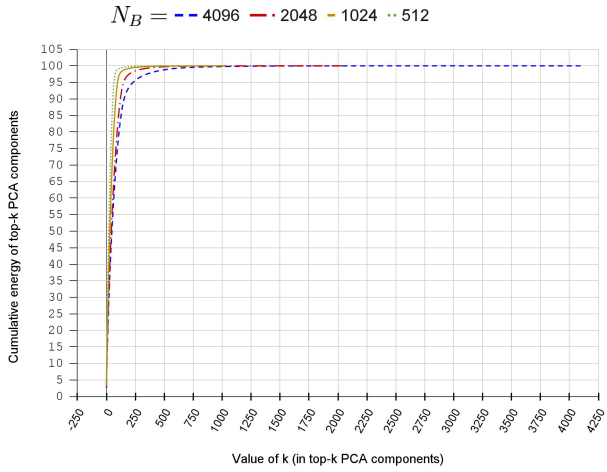

Figure 7: Cumulative energy of top-k PCA components for different values of Barlow projection layer's (Barlow projector's) output dimensionality.

we compute the answering accuracy for our COB model with different projector dimensions. We observe that projector corresponding to $N_B = 512$ yields the maximum answering accuracy, while the smaller value, i.e. $N_B = 256$, and the larger values, i.e. $N_B \geq 1024$, lead to inferior performances, which re-verifies our hyperparameter selection.

To select a good value of N_B we use a PCA analysis. We compute the cumulative energy of the top-k eigenvectors on a subset of VQA-CP v2 test set using our COB model for different values of projector dimension (N_B). From Figure 7, we observe that 512 eigenvectors contains at least 98.8% of total PCA energy for $N_B \leq 4096$. Hence, we chose $N_B = 512$ for all our experiments. Figure 7 supplements the Table 3 in the main paper.

Analysis of original Barlow-twins loss: In section 3.4.a of the main paper, we discuss that Barlow-twins [9] uses 1000 epochs of pre-training, suggesting a flatter loss curve. Here we plot the pre-training loss curve using the logs of the

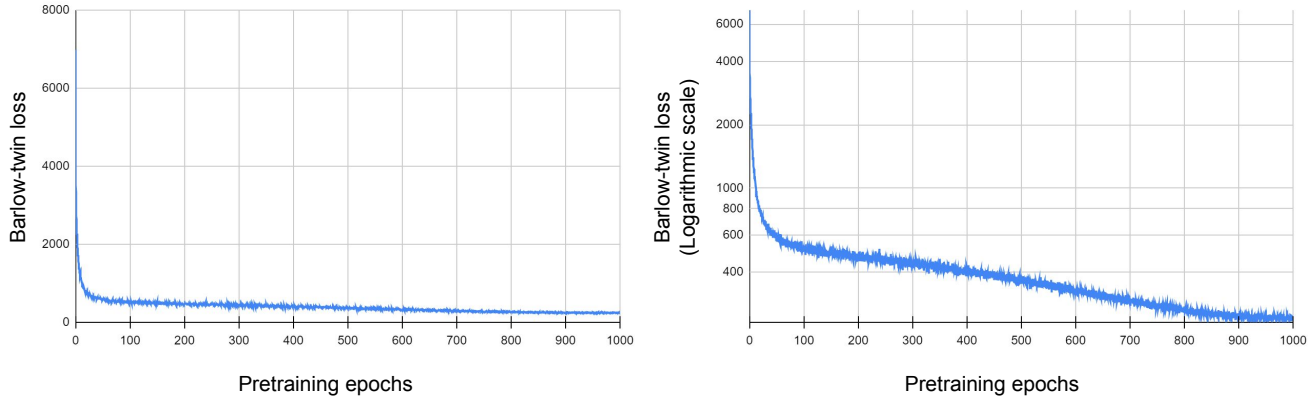

Figure 8: **Pre-training loss curve for official implementation of Barlow-twins [9]**: (Left) shows the decorrelation loss during pre-training for each epoch, (right) shows the decorrelation loss on a logarithmic scale for each epoch for a better visualization of the flatter region. We observe that Barlow-loss takes a longer gradient cycle to converge. **Note:** To plot these curves, we use the logs provided by the official implementation of Barlow-twins.

official implementation[1] of the Barlow-twins for reference, shown in Figure 8.

4. Additional Qualitative and Explainability results:

We provide more qualitative results in Figure 9, along with an additional set of inferior performing results in Figure 10. Similarly, more results on explainability using Grad-CAM [8] have been provided in Figure 11, along with additional set of results for the cases where COB performs similar or inferior to baseline GGE model in Figure 12.

4.1. Glossary

A glossary of all the mathematical notation used in the main paper and supplementary can be found in Table 2, 3.

Note: Figures and Tables are continued in the next pages.

[1]Official implementation of Barlow-Twins [9]: https://github.com/facebookresearch/barlowtwins

Figure 9: **More qualitative results:** Here we extend the qualitative result section of the main paper. Each of the image set/cell shows results for COB model (top-left), with top-5 prediction along with the probability scores corresponding to them, similarly bottom-left shows the GGE-DQ-iter baseline model prediction and bottom-right shows the top-5 baseline predictions with their probability scores. The ground truth answer is denoted by the answer with encapsulating green box. Red bounding box shows the maximal attention region in each image.

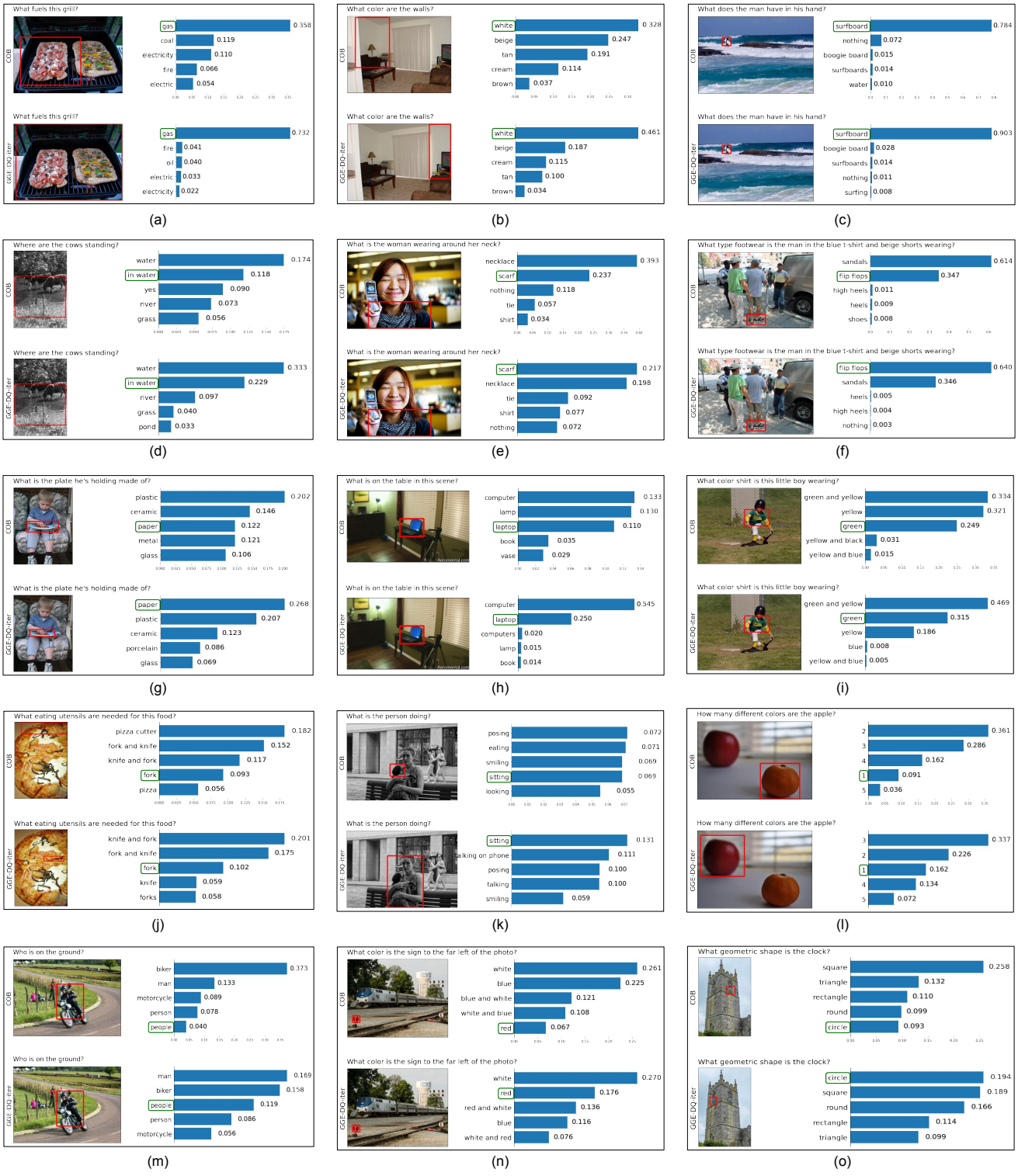

Figure 10: **Similar or Negative results w.r.t. baseline model:** Here we explicitly show the results where COB performs either equal or inferior to the baseline model. Each of the image set/cell shows results for COB model (top-left), with top-5 prediction along with the probability scores corresponding to them, similarly bottom-left shows the GGE-DQ-iter baseline model prediction and bottom-right shows the top-5 baseline predictions with their probability scores. The ground truth answer is denoted by the answer with encapsulating green box. Red bounding box shows the maximal attention region in each image.

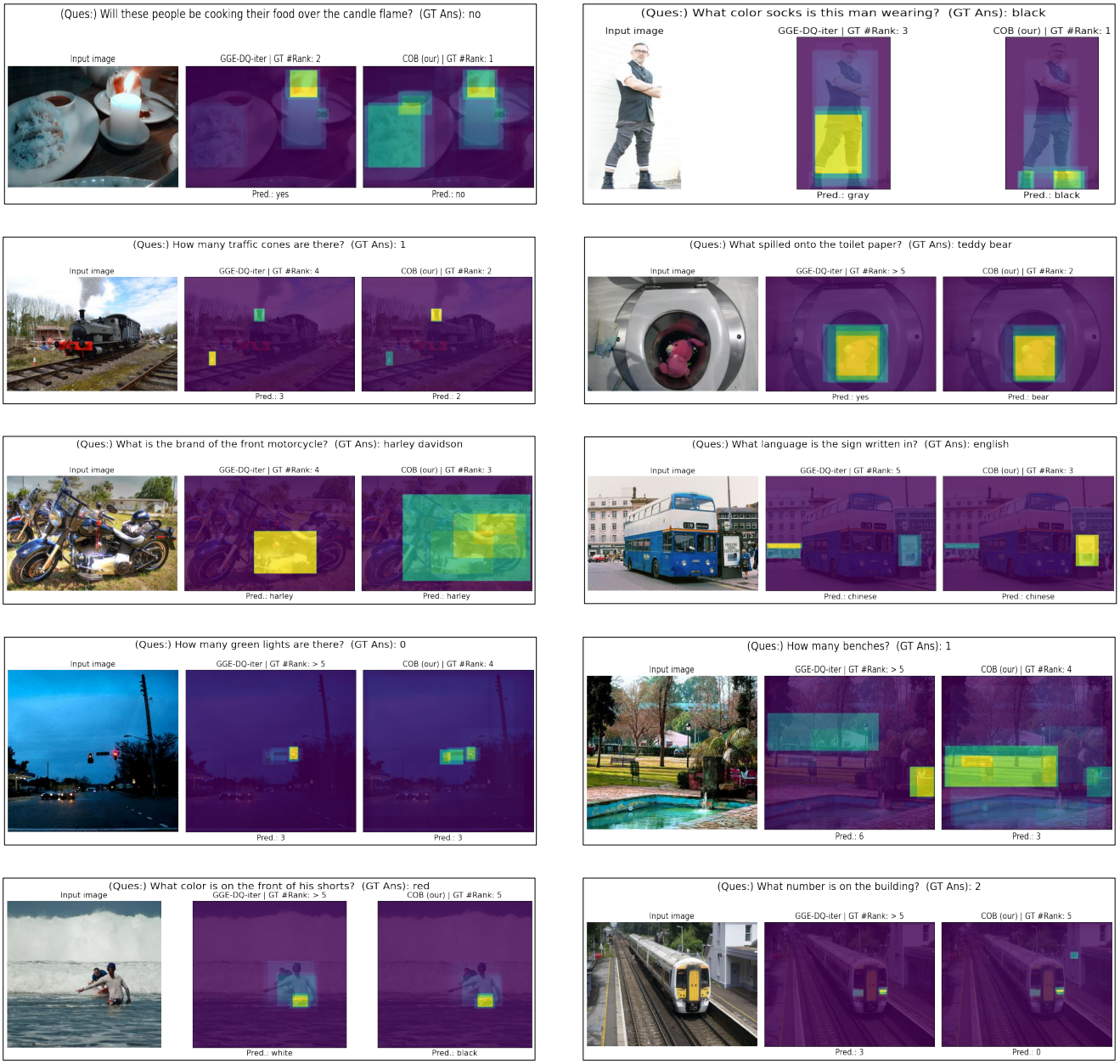

Figure 11: **More explainability results:** Here we extend the explainability results of the main paper. For each image set/cell: (top-text) is the input question along with the ground truth (GT) answer; left-image is the input image middle-image is the Grad-CAM [8] heatmaps computed by the baseline GGE-DQ-iter model overlaid on the original image; right-image is the overlaid Grad-CAM heatmap computed by COB; GT #Rank denotes the rank of the ground truth answer in the top-5 prediction by the respective models. 'Pred.' at the bottom of the middle and right images denotes the predicted answer with the highest probability score by the respective models.

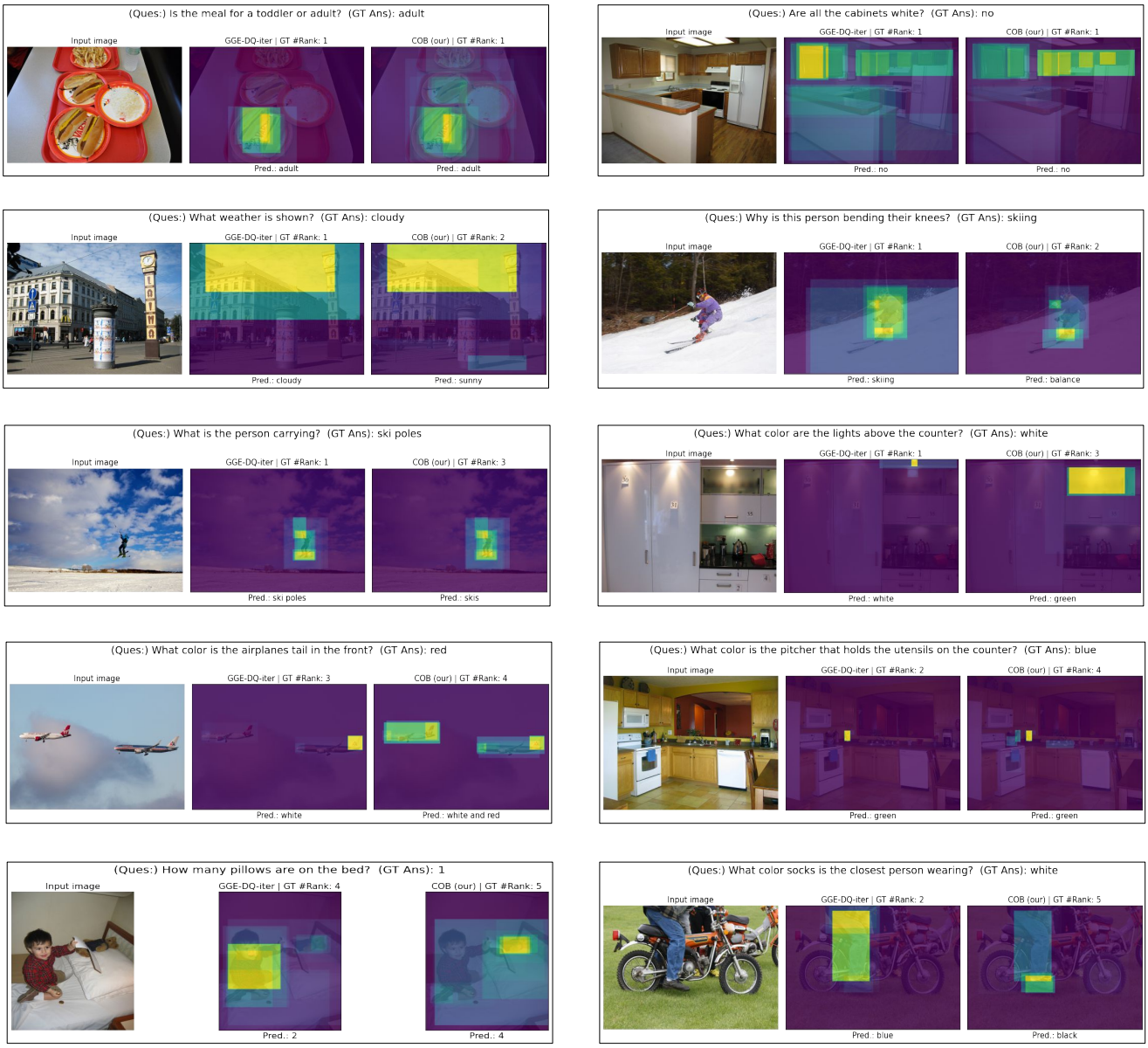

Figure 12: **Explainability results when COB performs similar or inferior to baseline (GGE-DQ-iter) model:** We observe that for the cases where the COB performs inferior to the baseline, the COB model still localizes either the same salient regions or better. This property of better salient localization also results in an improved CGD scores obtained by COB in comparison to all other state-of-the-art baselines, as discussed in the main paper. For each image set/cell: (top-text) is the input question along with the ground truth (GT) answer; left-image is the input image middle-image is the Grad-CAM [8] heatmaps computed by the baseline GGE-DQ-iter model overlaid on the original image; right-image is the overlaid Grad-CAM heatmap computed by COB; GT #Rank denotes the rank of the ground truth answer in the top-5 prediction by the respective models. 'Pred.' at the bottom of the middle and right images denotes the predicted answer with the highest probability score by the respective models.

Table 2: **Glossary of notations:** Definition of the notation use in the main manuscript and supplementary.

Notation	Meaning
$\mathcal{D}^{VQA}$	Distribution of input image question and answers.
$\mathcal{D}^{V}$	Distribution of input images.
$\mathcal{D}^{Q}$	Distribution of input question.
$\mathcal{D}^{A}$	Distribution of input answers.
d_k	An instance sampled from $\mathcal{D}^{VQA}$, indexed by k.
v_k	An instance sampled from $\mathcal{D}^{V}$.
q_k	An instance sampled from $\mathcal{D}^{Q}$.
a_k	An instance sampled from $\mathcal{D}^{A}$.
n_b	Number of samples in a mini-batch.
V	A mini-batch of n_b different instances (v_k) sampled from $\mathcal{D}^{V}$.
Q	A mini-batch of n_b different instances (q_k) sampled from $\mathcal{D}^{Q}$.
A	A mini-batch of n_b different instances (a_k) sampled from $\mathcal{D}^{A}$.
e_v	Pre-trained image encoder, parameters not updated during training.
e_q	Pre-trained language encoder, parameters not updated during training.
f_{θ_J}	Joint network with learnable parameters θ_J
m_k^f	A sample in the joint image+question embedding space.
M^f	A mini-batch of n_b different instances (m_k^f) sampled from $\mathcal{D}^{M}$.
$\mathcal{D}_M$	Distribution of samples (m_k^f) in the joint embedding space.
l_{θ_L}	A non-linear projection layer from joint embedding space to answer logit space.
m_k^l	Predicted answer logits.
M^l	A mini-batch consisting of n_b different instances of (m_k^l).
$\mathcal{L}_{CE}$	Cross-entropy loss, in general Categorical loss. For GGE [4], it is binary-cross entropy loss.
N_B	Dimensonality of space in which Barlow decorrelation loss is computed.
I	Identity matrix in real-value space ($\mathbb{R}$), of size ($N_B N_B$).
$\mathcal{D}^B \times \mathcal{D}^B$	A distribution space of matrices C computed between samples in $\mathcal{D}^B$.
$\mathcal{D}^S$	a modality specific distribution: For answers and joint representations, it is $\mathcal{D}^A$ and $\mathcal{D}^M$ respectively.
s_k	An instance sampled from $\mathcal{D}^S$.
S	A mini-batch of n_b different instances (s_k) sampled from $\mathcal{D}^S$.
e_s	Modality specific encoder. For questions, answers and images it is e_q, e_a and e_v respectively.

Notation	Meaning
$s_{k\|1}$ and $s_{k\|2}$	Two complementary samples sampled from $\mathcal{D}^S$, that makes a positive pair. In Barlow twin [9], these are two different augmentations of the same image.
s_k^s	Encoded representation of s_k using the encoder $e_s(.)$
S^s	A mini-batch consisting of n_b different instances of (s_k^s).
b_{θ_B}	A non-linear projector from encoded representation space $e_s(s_k)$ to Barlow space, parameterized by learnable parameters θ_B.
s_k^b	A non-linear projection of encoded representation $e_s(s_k)$ in the Barlow space.
S^b	A mini-batch consisting of n_b different instances of (s_k^b).
S_1 and S_2	Two complementary batches consisting of positive pairs, $s_{k\|1}$ and $s_{k\|2}$ for k samples in mini-batch.
S_1^b and S_2^b	Barlow projections of the two complementary batches, S_1, S_2.
$Norm(.)$	Batch normalization function [6].
$C(.)$	Correlation between two batches.
C^S	Correlation matrix between two complementary batches S_1^b and S_2^b.
i and j	i and j indexes the different feature components of the projected feature vector s_k^b.
C_{ij}^S	A single element of the correlation matrix C^S indexed by (i, j).
$\mathcal{L}_B^{\mathcal{S}}$	Barlow decorrelation loss for unimodal $\mathcal{D}^S$ input space.
e_a	Pre-trained language encoder, parameters not updated during training.
A^a	Encoded answer representation for the mini-batch A, using answer encoder $e_a(.)$.
V^v	Encoded image representation for the mini-batch A, using answer encoder $e_v(.)$.
Q^q	Encoded question representation for the mini-batch A, using answer encoder $e_q(.)$.
$b_{\theta_{B_M}}$	A non-linear projector from the joint representation space $M^l \in \mathcal{D}^M$ to Barlow space, parameterized by learnable parameters θ_{B_M}.
$C^{\mathcal{M}}$	Auto-correlation matrix computed on the barlow projection ($b_{\theta_{B_M}}(.)$) of the batch (M^l).
$b_{\theta_{B_A}}$	A non-linear projector from the encoded image representations A^a to Barlow space, parameterized by learnable parameters θ_{B_A}.
$C^{\mathcal{A}}$	Auto-correlation matrix computed on the barlow projection ($b_{\theta_{B_A}}(.)$) of the batch (A^a).
$C^{\mathcal{MA}}$	Cross-correlation matrix computed between barlow projected joint-representations and the encoded answer representations.
$\mathcal{L}_B^{\mathcal{O}}$	A Barlow decorrelation loss, where $\mathcal{O}$ denotes the input modalities.

Table 3: **Glossary of notations:** Definition of the notations used in the main manuscript and supplementary. (Continuation of table 2.)

Notation	Meaning
$\mathcal{L}_B^{\mathcal{M}}$	A unimodal Barlow decorrelation loss for joint image-question embedding space ($\mathcal{D}^M$).
$\mathcal{L}_B^{\mathcal{A}}$	A unimodal Barlow decorrelation loss for answer space ($\mathcal{D}^A$).
$\mathcal{L}_B^{\mathcal{M}\mathcal{A}}$	A multimodal Barlow decorrelation loss between the joint image-question embedding space ($\mathcal{D}^M$) and answer space ($\mathcal{D}^A$).
$\mathcal{L}_B$	Overall Barlow decorrelation loss.
$\mathcal{L}_{all_{base}}$	Baseline (naive) implementation of overall (categorical + Barlow decorrelation) losses.
n	Number of pre-training epochs (with categorical loss $\mathcal{L}_{CE}$) before applying Barlow decorrelatin loss ($\mathcal{L}_B$), in Align then Barlow (ATB) formulation.
$\mathcal{L}_{all_{ATB}}$	Overall loss formulation for ATB training policy.
$\mathcal{L}_{all_{COB}}$	Our overall constrained optimization formulation (i.e. categorical loss constrained with Barlow (COB) decorrelation loss formulation).

Notation	Meaning
λ	A learnable Lagrange multiplier to weight categorical loss $\mathcal{L}_{CE}$ and Barlow decorrelation loss $\mathcal{L}_B$.
κ	It is a tolerance hyperparameter to control the change in λ, give the value of Barlow decorrelation loss $\mathcal{L}_B$ at iteration t.
$\mathbb{C}$	Barlow constraint defined as difference between $\mathcal{L}_B$ and κ. When it becomes zero the change in λ becomes negative. This subsequently forces the dynamic weight λ assigned to constraint to decrease.
$\mathbb{C}_t$	Constraint $\mathbb{C}$ at iteration t.
λ_t	value of λ at iteration t.
$\mathcal{L}_{all_{COB^\lambda}}$	Lagrangian form of overall constrained optimization $\mathcal{L}_{all_{COB}}$.
$\triangle\lambda_t$	Change in λ in iteration t.

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018. 2

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 1, 2

[3] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 2

[4] Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. Greedy gradient ensemble for robust visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1584–1593, 2021. 1, 10

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 10

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2

[8] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5, 8, 9

[9] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 4, 5, 10