
(a) Structural Causal Model (b) The definition of $do(X)$

Figure 1: **Illustration of Structural Causal Model (SCM) and *do*-calculus definition**.

# A Appendix

## A.1 Additional Preliminaries

### A.1.1 Causal Inference

**Structural Causal Model (SCM).** SCM is a statistical model representing the causal relationship between variables in the graph structure Glymour et al. (2016). In the causal graph, each variable is denoted by nodes, and 'causation' between two different variables is denoted by a directed edge between nodes. Fig. 1a illustrates an example of SCM representation in graph form. The edge $X \rightarrow Y$ in the graph implies that $X$ is the 'cause' of $Y$. Also, $Z$ in the graph represents a 'confounder', which simultaneously affects both $X$ and $Y$, therefore making it difficult to find a true effect of $X$ on $Y$. Such confounder induces the spurious correlation between $X$ and $Y$ through the backdoor path between $X$ and $Y$. A backdoor path is formally defined as any path from $X$ to $Y$ that starts with an arrow pointing to $X$ (Yang et al., 2021), such as $X \leftarrow Z \rightarrow Y$ in 1a. To find out the true causal relationship between $X$ and $Y$, the causal intervention with *do*-calculus $P(Y|do(X))$ is applied to cut-off the relationship $Z \rightarrow X$, as illustrated in 1b, therefore removing the spurious correlation induced by $Z$. Backdoor adjustment is a widely adopted approach to deconfound the effect of the confounders $Z$ using the *do*-calculus, which we further concretize in the very following section.

**The Backdoor Adjustment.** Given a directed acyclic graph consisting of $X$, $Y$, and $Z$ as in 1a, backdoor adjustment can be applied to reveal the true causal effect of the $X$ on $Y$ given the confounder $Z$. By Bayes' theorem, $P(Y|X)$ can be expressed as follows:

$$P(Y|X) = \sum_{z \in Z} P(Y|X, Z = z)P(Z = z|X). \tag{1}$$

The causal intervention with *do*-calculus $P(Y|do(X))$ mentioned in the previous section is then formally defined as below:

$$P(Y|do(X)) = \sum_{z \in Z} P(Y|X, Z = z)P(Z = z). \tag{2}$$

Through the backdoor adjustment, the true causal relationship between $X$ and $Y$, which is denoted as $P(Y|do(X))$ is measured without any effect of the confounder $Z$.

**Normalized Weighted Geometric Mean (NWGM).** To approximate $P(Y|do(X))$, we use NWGM. Before dealing with NWGM, we first revisit the definition of Weighted Geometric Mean (WGM). Given a discrete variable X and its distribution $P(X)$, the expectation of $f(x)$ is defined as:

$$\mathbb{E}_x[f(x)] = \sum_{x \in X} f(x)P(x). \tag{3}$$

The Weighted Geometric Mean (WGM), an approximation of $\mathbb{E}_x[f(x)]$ is defined as follows:

$$WGM(f(x)) = \prod_{x \in X} f(x)^{P(x)}. \tag{4}$$

If the activation function of $f(x)$ is a composition of a function $g(x)$ followed by an exponential function, *i.e.*, $f(x) = \exp(g(x))$, Eq. 4 can be reformulated as:

$$\begin{aligned} WGM(f(x)) &= \prod_{x \in X} \exp[g(x)]^{P(x)} = \prod_{x \in X} \exp[g(x)P(x)] \\ &= \exp(\sum_{x \in X} g(x)P(x)) = \exp\{\mathbb{E}_x[g(x)]\}. \end{aligned} \tag{5}$$

Interpreting WGM in the perspective of deep learning, $f(x)$ can be regarded as a neural network whose last activation function is the softmax function. Therefore, Xu et al. (2015) and Yang et al. (2021) approximate the expectation of the $f(x)$ using the WGM as follows:

$$\mathbb{E}_x[f(x)] \approx WGM(f(x)) = \exp\{\mathbb{E}_x[g(x)]\} \tag{6}$$

To guarantee that output logits can be interpreted as a probability, NWGM, a normalized version of WGM, is applied so that the sum of output logits adds up to one, and it is formally defined as:

$$\begin{aligned} NWGM(f(x)) &= \frac{\prod_x \exp(g(x))^{P(x)}}{\sum_j \prod_x \exp(g(x))^{P(x)}} \\ &= \frac{\exp(\mathbb{E}_x[f(x)])}{\sum_j \exp(\mathbb{E}_x[f(x)])} \\ &= Softmax(\mathbb{E}_x[f(x)]) \end{aligned} \tag{7}$$

Adopting the WGM defined above to our model, $P(Y|do(X))$ can be approximated as below, where $P(Y|X,z) = Softmax(g(X,z)) \propto \exp(g(X,z))$:

$$\begin{aligned} P(Y|do(X)) &= \mathbb{E}_z[P(Y|X,z)] \\ &= \mathbb{E}_z[\exp(g(X,z))] \\ &\approx \exp(\mathbb{E}_z[g(X,z)]) \\ &= \exp\{\sum_{z \in Z} (f(X) + z + \tilde{z})\, P(z)\}. \end{aligned} \tag{8}$$

where $g(X,z) = f(X) + z + \tilde{z}$. Then, we apply NWGM to normalize Eq. 8 as to get final deconfounded prediction probabilities $P(Y|do(X))$ as follows:

$$\begin{aligned} P(Y|do(X)) &\approx Softmax(\mathbb{E}_z[g(X,z)]) \\ &= Softmax\{\sum_{z \in Z} (f(X) + z + \tilde{z})\, P(z)\}. \end{aligned} \tag{9}$$

### A.1.2 Generalized Cross Entropy (GCE) Loss

**GCE.** GCE loss was first proposed as a generalized loss taking advantage of both Mean Absolute Error (MAE) loss, and Categorical Cross Entropy (CCE) loss by Zhang & Sabuncu (2018). Given an input $x$, the ground truth one-hot vector $y$, and the set of parameters $\theta$ of the classifier $f$, MAE and CCE loss are formally defined as below in the common case where the softmax is followed by the classification layer:

$$\begin{aligned} \mathcal{L}_{MAE}(f(x;\theta), y) &= ||y - f(x;\theta)||_1 \\ \mathcal{L}_{CCE}(f(x;\theta), y) &= -\sum_{j=1}^{C} y_j \log f_j(x;\theta), \end{aligned} \tag{10}$$

where $C$ denotes the number of target classes, $y_j$ and $f_j$ denote the $j$-th element of $y$ and the $j$-th prediction of $f$. The gradient of loss functions with respect to parameter $\theta$ is as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_{MAE}(f(x;\theta), y)}{\partial \theta} &= -\nabla_\theta f_y(x;\theta) \\ \frac{\partial \mathcal{L}_{CCE}(f(x;\theta), y)}{\partial \theta} &= -\frac{1}{f_y(x;\theta)} \nabla_\theta f_y(x;\theta), \end{aligned} \tag{11}$$

where $f_y$ denotes the element of the output logit corresponding to the ground-truth label. As formulated in Eq. 11, CCE emphasizes samples with larger $1/f_y(x;\theta)$, or smaller $f_y(x;\theta)$. On the contrary, MAE equally treats every sample with the same weight. The fact that MAE does not place a larger weight on difficult samples makes MAE robust to noisy labels, but it also makes training difficult since every sample is treated equally so that challenging examples are not learned enough. In contrast, optimizing a model using CCE is easier due to larger weights being given to challenging samples. However, CCE is sensitive to noisy labels, since the model could easily be overfitted to such noisy samples which are intrinsically difficult due to label noise. Then GCE loss can be viewed as a generalization between MAE and CCE loss, and is formally defined as below:

$$\mathcal{L}_{GCE}(f(x;\theta),y) = \frac{1-p_y(x;\theta)^q}{q}, \tag{12}$$

where $q \in (0,1]$ is a smoothing parameter. The gradient of $\mathcal{L}_{GCE}$ with respect to $\theta$ is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{GCE}(f(x;\theta),y)}{\partial\theta} &= f_y(x;\theta)^q\left(-\frac{1}{f_y(x;\theta)}\nabla_\theta f_y(x;\theta)\right) = f_y(x;\theta)^q\frac{\partial\mathcal{L}_{CCE}}{\partial\theta} \\ &= -f_y(x;\theta)^{q-1}\nabla_\theta f_y(x;\theta) = f_y(x;\theta)^{q-1}\frac{\partial\mathcal{L}_{MAE}}{\partial\theta}.\end{aligned} \tag{13}$$

Therefore, $\mathcal{L}_{GCE}$ additionally weights each sample by $f_y(x;\theta)^q$ times compared to CCE loss, weighting difficult samples less. Also, it weights each sample by $f_y(x;\theta)^{q-1}$ times compared to MAE loss, giving larger weight to difficult examples compared to MAE loss. If $q$ is properly chosen, GCE can therefore act as a generalized loss that is more robust than CCE and easier to train than MAE, achieving a balanced trade-off between two losses.

**GCE Loss in Computer Vision.** By the fact that GCE loss gives smaller weights to 'difficult' examples compared to conventional CCE loss, Lee et al. (2021) and Nam et al. (2020) propose capturing bias in the model by leveraging GCE loss to train a 'biased network', which is overfitted to easy samples, which corresponds to 'bias' or 'spurious correlation' existing in the dataset. Both works train the model with GCE loss to achieve the model to be biased by focusing on the "easier" samples compared to the conventional CCE.

### A.2 Experimental Settings

#### A.2.1 Dataset

We validate the proposed model on four benchmark datasets: **TGIF-QA** (Li et al., 2016; Jang et al., 2017), **MSVD-QA** (Chen & Dolan, 2011; Xu et al., 2017), and **MSRVTT-QA** (Xu et al., 2016; 2017). **TGIF-QA** consists of 103,913 QA pairs from 56,720 GIFs and includes three multiple-choice VideoQA tasks: repetition count, repeating action, and state transition, along with an open-ended frameQA task reasoning on a single frame. **MSVD-QA** and **MSRVTT-QA** are both open-ended VideoQA datasets with descriptive QA tasks, while **MSRVTT-QA** consists of more complex and longer 10,000 trimmed videos and larger 243,000 QA pairs compared to **MSVD-QA** with 1,970 trimmed videos and 50,500 QA pairs.

#### A.2.2 Implementation Details.

**Model Architecture.** We adopt the Transformer (Vaswani et al., 2017) architecture with 12 layers for both the data encoder $f$ and the confounder encoder $g$. Concretely, for the data encoder $f$, visual tokens $X_v$ and text tokens $X_q$ are concatenated with an additional [CLS] token to form an input $X = (x_q, x_v) \in \mathbb{R}^{N\times D}$. To build $x_v$, we sample 3 frames per single input video. Each frame has a spatial resolution of 224×224, and is patchified into 14×14 patches with the size of 16×16 for each. For text token $x_q$, we set 40 as the max length of the input text sequence. An input text is then tokenized to have a hidden dimension of $D = 768$. After concatenating $x_q$ and $x_v$, modality encoding is added to input tokens having corresponding modalities. When conducting cross-attention in $g$, we apply a stop-gradient operation to $\tilde{X}$ so that it could not be affected by $\mathcal{L}_{\text{confounder}}$. Also, we use $M = 128$ for the number of confounder query tokens.

**Algorithm 1** Overall Algorithm

**Inputs:** sample $\{X = (x_q, x_v), Y\}$, negative sample $\{X' = (x'_q, x'_v), Y'\}$, confounder queries $Z$, number of confounder queries $M$
**Parameters:** prior probability $c$, data encoder $f$, confounder encoder $g$, FFN $\{h_f, h_g\}$

$X_q,\ X_v \leftarrow (x_q, x'_v),\ (x'_q, x_v)$
$\tilde{X},\ \tilde{X}_q,\ \tilde{X}_v \leftarrow f(X),\ f(X_q),\ f(X_v)$
$\tilde{Z}, \tilde{Z}_q, \tilde{Z}_v \leftarrow \{\tilde{z}|\tilde{z} = g(\tilde{X}, z), \forall z \in Z\}, \{\tilde{z}|\tilde{z} = g(\tilde{X}_q, z), \forall z \in Z\}, \{\tilde{z}|\tilde{z} = g(\tilde{X}_v, z), \forall z \in Z\}$
$\hat{Y}_f \leftarrow h_f\left(\sum_{z \in Z}\left(\tilde{X} + z + \tilde{z}\right) c_z\right)$ ▷ $c_z$ is a prior probability of $z$
$\tilde{Z}_{q,q},\ \tilde{Z}_{q,v},\ \tilde{Z}_{v,q},\ \tilde{Z}_{v,v} \leftarrow \tilde{Z}_q[0:M/2],\ \tilde{Z}_q[M/2:M],\ \tilde{Z}_v[0:M/2],\ \tilde{Z}_v[M/2:M]$
$\hat{Y}_g^{(q,q)},\ \hat{Y}_g^{(q,v)},\ \hat{Y}_g^{(v,q)},\ \hat{Y}_g^{(v,v)} \leftarrow h_g(\tilde{Z}_{q,q}),\ h_g(\tilde{Z}_{q,v}),\ h_g(\tilde{Z}_{v,q}),\ h_g(\tilde{Z}_{v,v})$
$\mathcal{L}_{\text{causal}} \leftarrow \text{CE}(\hat{Y}_f, Y)$
$\mathcal{L}_{\text{confounder}} \leftarrow \text{GCE}(\hat{Y}_g^{(q,q)}, Y) + \text{GCE}(\hat{Y}_g^{(q,v)}, Y') + \text{GCE}(\hat{Y}_g^{(v,q)}, Y') + \text{GCE}(\hat{Y}_g^{(v,v)}, Y)$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{causal}} + \mathcal{L}_{\text{confounder}}$
**return** $\mathcal{L}$

**Training Details.** For training, the initial learning rate is set to $10^{-4}$ with cosine decay and warmup applied until 10% of the total training step is done. We train the models with AdamW (Loshchilov & Hutter, 2017) optimizer with a weight decay rate of 0.01. The probability of confounder dropout is 0.15. Our backbone encoders are pretrained on Webvid (Bain et al., 2021), YT-Temporal 180M (Zellers et al., 2021), HowTo100M (Miech et al., 2019), CC3M (Sharma et al., 2018), CC12M (Changpinyo et al., 2021), COCO (Lin et al., 2014), VisualGenome (Krishna et al., 2017), and SBU (Ordonez et al., 2011) as in Fu et al. (2021) and Wang et al. (2022). All the experiments are conducted on 4 $\times$ Tesla A100 GPUs.

**MCQA Details.** We concatenate each option and the question and insert the [SEP] token between them to construct the text token sequence. To efficiently calculate $\mathcal{L}_{\text{confounder}}$, we only take into account two negative pairs $(X_q, Y)$ and $(X_v, Y')$ instead of four negative pairs including $(X_q, Y')$ and $(X_v, Y)$, *i.e.*, $\mathcal{L}_{\text{confounder}} = \text{GCE}(\hat{Y}_g^{(q,q)}, Y) + \text{GCE}(\hat{Y}_g^{(v,q)}, Y')$. This is because it is cumbersome to forward all the combinations of negative pairs including concatenated text token sequences for each option.

### A.3 Overall Algorithm

The overall algorithm to train our proposed framework is formulated in Alg. 1.

## References


Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. In *NeurIPS*, 2021.

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *CVPR*, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, 2020.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

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

Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACMMM*, 2017.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *CVPR*, 2021.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. In *NeurIPS*, 2021.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.