1. Derivation of NWGM Approximation

We will show the derivation of Normalized Weighted Geometric Mean (NWGM) approximation used in Eq.(7) in the main paper. In a MIL problem, given the bag feature, let \( g(\cdot) \) be a classifier that calculates logits for the \( k \)-way bag-level classification. The approximation moves the outer sum into softmax: \( \mathbb{E}[\text{Softmax}(g(\cdot))] \approx \text{Softmax}(\mathbb{E}[g(\cdot)]) \). Without loss of generality, the backdoor adjustment formula in Eq.(3) can be written as:

\[
P(Y = y \mid \text{do}(X = x)) = \sum_{c \in C} \text{Softmax}(g_y(x \oplus h)) P(c),
\]

where \( C \) denotes the confounder stratifications, \( g_y \) is the classifier logit for class \( y \), \( h = h(x, c) \) is the feature concatenated to \( x \) (see Eq.(6) in the main paper), and \( P(c) \) is the prior for each confounder. Then, the NWGM of Eq. (A1) can be achieved as:

\[
\sum_{c \in C} \text{Softmax}(g_y(x \oplus h)) P(c) \approx \text{NWGM}_{c \in C} (\text{Softmax}(g_y(x \oplus h)))
\]

(A3)

\[
= \frac{\prod_{c} \left( \exp (g_y(x \oplus h)) \right)^{P(c)}}{\sum_{c} \left( \exp (g_y(x \oplus h)) \right)^{P(c)}}
\]

(A4)

\[
= \frac{\exp \left( \sum_{c} g_y(x \oplus h) P(c) \right)}{\sum_{c} \exp \left( \sum_{c} g_y(x \oplus h) P(c) \right)}
\]

(A5)

\[
= \text{Softmax} \left( \mathbb{E}_c [g_y(x \oplus h)] \right),
\]

(A6)

where Eq. (A3) follows [1] and Eq. (A4) is the definition of NWGM. Note that we set \( g(\cdot) \) be a linear classifier by default, which can be written as \( g(x \oplus h) = H_1 x + H_2 h \), where \( H_1, H_2 \in \mathbb{R}^{k \times d} \) are learnable weight, \( d \) is the dimension of \( x \) and \( h \). Then, in Eq. (A6), the term inner Softmax can be written as:

\[
\sum_{c} g(x \oplus h) P(c) = \sum_{c} (H_1 x + H_2 h) P(c)
\]

(A7)

\[
= H_1 x + \sum_{c} H_2 h P(c)
\]

(A8)

\[
= g \left( x \oplus \sum_{c} h P(c) \right)
\]

(A9)

where Eq. (A8) is because the feature of \( x \) is same for all confounder \( c \), and we can discard the \( \mathbb{E} \) over \( x \). Putting Eq. (A9) into Eq. (A6), we can get the Eq.(7) in the main paper.

2. More details about feature extractors

Generally, we adopt ResNet18 [5], ViT-small [4], and CTransPath [12] as feature extractors respectively.

**ResNet-18** is a basic and widely-used CNN model in the community of WSIs. We adopt the ImageNet pretrained model officially released by PyTorch (https://download.pytorch.org/models/resnet18-5c106cde.pth). For each instance, ResNet-18 outputs the feature of 512 dimension from the penultimate layer. **ViT-small** is a typical transformer-based model. We build upon the visual transformer architecture from [10] based on the timm library [13]. We adopt the model pretrained with MoCo V3’s manner [2]. For each instance, ViT-small outputs the class token, which is of 384 dimension. **CTransPath** is hybrid CNN-transformer feature extractor, which combines the ResNet structure and Swin Transformer blocks [7]. We adopt the model pretrained with a semantically-relevant contrastive learning (SRCL) manner [12], where the positives include augmentation views and multiple similar ones from memory bank (measured by cosine similarity metric). For each instance, CTransPath outputs the feature of 768 dimension from average pooling layer. For ViT-small and CTransPath, they are self-supervised pre-trained on 9 datasets: UniToPatho, TissueNet, NCT-CRC-HE, Colorectal cancer, Camelyon16, TCGA-NSCLC, TCGA-RCC, MIDOG, and CRAG, containing around 15 million unlabeled patches. The pretrained ViT and CTransPath can be downloaded from https://github.com/Xiyue-Wang/TransPath.

3. More details about aggregators

We use DSMIL’s code base for implementation and evaluation, and build other models based on their officially released codes.

- The official code for ABMIL can be referred to https://github.com/AMLab-Amsterdam/
AttentionDeepMIL.

- The official code for DSMIL can be referred to https://github.com/binli123/dsmil-wsi.

- The official code for TransMIL can be referred to https://github.com/szc19990412/TransMIL.

- The official code for DTFD-MIL can be referred to https://github.com/hrzhang1123/DTFD-MIL.

Following their codes, we use the Adam optimizer for ABMIL and DSMIL with the cosine decay schedule [8]. The bag feature is the attention-weighted sum of instance features. For TransMIL, Lookahead optimizer [15] is employed with a weight decay of 1e-5. The bag feature is represented by the class token. We use the Adam optimizer for DFTD-MIL with MultiStepLR schedule. The bag feature is generated by Tier-2. The Interventional training of stage 3 can be referred to Algorithm 1.

Algorithm 1 Pseudocode of Interventional Training

```python
# Inputs:
# Confounder dictionary C with shape (K, d),
# Features of instances [b_1, ..., b_n], each
# with shape (1, d)
# Ground truth Y

# Outputs:
# Bag level prediction Y_hat, Loss L for
# optimizing network parameters

# Previous MIL training:
# B = aggregator_network([b_1, ..., b_n]) # B is
# the bag feature with shape (1, d)
# Y_hat = classify_head(B)
# L = criterion(Y_hat, Y)

# Interventions training:
# B = aggregator_network([b_1, ..., b_n]) # B is
# the bag feature with shape (1, d)
# B_q = linear1(B) # Projection matrix W_1, B_q
# with shape (1, l)
# C requires_grad = False # freeze C
# C_k = linear2(C) # Projection matrix W_2, C_k
# with shape (K, 1)
# Alpha = torch.mm(C_k, B_q.T)
# Alpha = F.softmax(Alpha / sqrt(1), dim=0) #
# Normalize weighted scores
# C_ave = torch.mm(Alpha, T, C) # Weighted average,
# C_ave with shape (1, d)
# B = torch.cat([B, C_ave], dim=1)
# Y_hat = classify_head(B)
# L = criterion(Y_hat, Y)
```

4. More results about DTFD-MIL (MaxMinS)

To further verify the effectiveness of IBMIL with baseline of DTFD-MIL, we switch to “MaxMinS” as the feature distillation strategy, and provide the results in Tab. 1 and Tab. 2. As can be seen, IBMIL can bring consistent performance boost under different feature distillation strategies, which demonstrates the effectiveness of our proposed scheme.

5. More Baselines

Comparison with IMIL. IMIL [6] applies instance-level physical intervention (i.e., MoCo V2 style augmentation) for robust instance label prediction, while IBMIL is based on the backdoor adjustment for bag label prediction. To compare with IMIL, we apply instance-level physical intervention for bag label prediction. As shown in Tab. 3, the results (i.e., ABMIL+IMIL) are even worse than baseline
Table 3. The performance with ImageNet pre-trained Res-18.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PRE</th>
<th>REC</th>
<th>ACC</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABMIL</td>
<td>86.71</td>
<td>81.71</td>
<td>84.50</td>
<td>84.07</td>
</tr>
<tr>
<td>ABMIL+IMIL</td>
<td>76.24</td>
<td>73.00</td>
<td>75.97</td>
<td>74.60</td>
</tr>
<tr>
<td>ABMIL+IBMIL</td>
<td>88.58</td>
<td>87.14</td>
<td>88.37</td>
<td>90.43</td>
</tr>
<tr>
<td>ABMIL+ColorNorm [3]</td>
<td>83.02</td>
<td>79.76</td>
<td>82.17</td>
<td>85.59</td>
</tr>
</tbody>
</table>

Figure 1. (a) T-SNE visualization. (b) Attention maps.

on Camelyon16, since the strong augmentation achieves the causal intervention at the cost of affecting the statistical information in the bag.

Color as A confounder and Experimental Setup. In computational pathology, stain color variation is a common issue causing generalization error. From the causal lens, color is a kind of bag contextual confounders causing spurious correlations between bags and labels. Thorough evaluation was conducted on patch-based classification to highlight this issue and it claims that the conclusions generalize to WSI classification as well [9]. We conduct experiments with color normalization [3], the results (i.e., ABMIL+ColorNorm) in Tab. 3 achieve better AUC than baseline. Note that IBMIL still outperforms it as there exist other confounders in general cases.

Relations to ReMix [14]. Clustering is used in ReMix and our work but with different implementations and purposes. In ReMix, clustering is performed at patch-level for each bag, and the prototypes are used to represent the bag. In our method, clustering is performed at bag-level, and the prototypes are used to approximate the confounders for backdoor adjustment.

6. Qualitative Analysis

T-SNE. In Fig. 1a, we visualize the bag features via t-SNE and denote the prototypes by stars. We empirically find that color is abstracted in some clusters. Note that confounders can be any bag contextual information (e.g., color, texture or patient-specific patterns). Lacking these attribute labels hinders us from further analysis. Thus, we will turn to expert pathologist knowledge for further exploration.

Attention Map. IBMIL is proposed to empower existing bag MIL methods generally (including non-parametric ones), thus no explicit constraints are applied to attention. In Fig. 1b, the attention maps are achieved by subtraction and binarization between IBMIL and baseline, and we find IBMIL pays more attention in tumoral regions in some cases. A potential improvement is to further incorporate attention-based interventions [11].

References

