

# Channel Correlation Loss for Binary Neural Networks

Xindi Zuo Wei Zhang\* Hai Yu Zhiliang Zhu  
Northeastern University, China

zuoxd@mails.neu.edu.cn zhangwei@swc.neu.edu.cn {yuh, zzl}@mail.neu.edu.cn

## Section A: Gradient Upper Bound Derivation for CC-Loss

We begin by outlining the structure of the total loss function and decomposing it. The total loss is defined as:

$$\mathcal{L}_{CC} = \frac{1}{B} \sum_{b=1}^B \mathcal{L}_{CC}^{(b)}$$

where  $B$  is the batch size, and  $\mathcal{L}_{CC}^{(b)}$  is the loss for the  $b$ -th sample. The loss for a single sample is:

$$\mathcal{L}_{CC}^{(b)} = \log \left( 1 + \frac{\beta}{\tau} S_b \right)$$

where  $\beta$  and  $\tau$  are tuning parameters, and  $S_b$  is the energy term associated with the  $b$ -th sample.

The energy term is computed as:

$$S_b = \sum_{i=1}^C \sum_{j=i+1}^C \max \left( 0, \tilde{\mathbf{z}}_i^{(b)\top} \tilde{\mathbf{z}}_j^{(b)} \right)^2$$

where  $\tilde{\mathbf{z}}_i^{(b)}$  represents the  $i$ -th feature of the  $b$ -th sample, normalized such that  $\|\tilde{\mathbf{z}}_i^{(b)}\| = 1$ .

Next, we compute the gradient of the loss with respect to the energy term  $S_b$ :

$$\frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial S_b} = \frac{\beta/\tau}{1 + \frac{\beta}{\tau} S_b}$$

An important result from this computation is that the sensitivity of the loss to the energy term is bounded:

$$\left| \frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial S_b} \right| \leq \frac{\beta}{\tau}$$

which ensures that the gradient does not explode.

We then consider the part of  $S_b$  that is dependent on  $\tilde{\mathbf{z}}_i$ . The gradient of  $S_b$  with respect to  $\tilde{\mathbf{z}}_i$  is given by:

$$\frac{\partial S_b}{\partial \tilde{\mathbf{z}}_i} = \sum_{j \neq i} 2 \cdot \mathbf{1}_{\{\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j > 0\}} (\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j) \tilde{\mathbf{z}}_j$$

Next, using the chain rule, we combine the results to compute the gradient of the loss with respect to  $\tilde{\mathbf{z}}_i$ :

$$\frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial \tilde{\mathbf{z}}_i} = \frac{\beta/\tau}{1 + \frac{\beta}{\tau} S_b} \cdot \sum_{j \neq i} 2 \cdot \mathbf{1}_{\{\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j > 0\}} (\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j) \tilde{\mathbf{z}}_j$$

We then apply the triangle inequality to bound the gradient's norm:

$$\left\| \frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial \tilde{\mathbf{z}}_i} \right\| \leq \left\| \frac{\beta/\tau}{1 + \frac{\beta}{\tau} S_b} \right\| \cdot \left\| \sum_{j \neq i} 2 \cdot \mathbf{1}_{\{\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j > 0\}} (\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j) \tilde{\mathbf{z}}_j \right\|$$

By further bounding the norm of the inner sum, we obtain:

$$\left\| \sum_{j \neq i} 2 \cdot \mathbf{1}_{\{\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j > 0\}} (\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j) \tilde{\mathbf{z}}_j \right\| \leq 2(C-1)$$

Finally, combining the bounds on the outer term, we obtain the following upper bound for the gradient:

$$\left\| \frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial \tilde{\mathbf{z}}_i} \right\| \leq \frac{2\beta(C-1)}{\tau}$$

Extending this result to the batch gradient, we have:

$$\frac{\partial \mathcal{L}_{CC}}{\partial \tilde{\mathbf{z}}_i} = \frac{1}{B} \sum_{b=1}^B \frac{\partial \mathcal{L}_{CC}^{(b)}}{\partial \tilde{\mathbf{z}}_i}$$

Applying the triangle inequality, we get the final upper bound for the batch gradient:

$$\left\| \frac{\partial \mathcal{L}_{CC}}{\partial \tilde{\mathbf{z}}_i} \right\| \leq \frac{2\beta(C-1)}{\tau}$$

## Section B: Detailed Theory of Three Variants

In our ablation studies, we systematically compared three distinct approaches to channel decorrelation. Below, we provide the formal mathematical formulations and theoretical insights for each variant.

\*Corresponding author.

## 1. Statistical Approach (Covariance Minimization)

The statistical approach aims to minimize the off-diagonal elements of the feature covariance matrix, promoting statistical independence among channels.

**Formal Definition:**

$$\mathcal{L}_{\text{stat}} = \frac{1}{C^2} \|\Sigma - \text{diag}(\Sigma)\|_F^2 \quad (1)$$

where the covariance matrix  $\Sigma$  is computed as:

$$\Sigma = \frac{1}{B} \sum_{b=1}^B (\mathbf{Z}_b - \mu)(\mathbf{Z}_b - \mu)^T$$

with  $\mathbf{Z}_b \in \mathbb{R}^{C \times HW}$  representing the feature matrix for sample  $b$ , and  $\mu = \frac{1}{B} \sum_{b=1}^B \mathbf{Z}_b$  being the batch mean.

**Theoretical Basis:** This method is grounded in the principle that uncorrelated features facilitate more efficient representations. By minimizing the Frobenius norm of the off-diagonal covariance elements, we enforce approximate statistical independence:

$$\mathbb{E}[\mathbf{z}_i \mathbf{z}_j^T] \approx 0 \quad \forall i \neq j$$

**Limitations in BNN Context:**

- **Magnitude Sensitivity:** Covariance computation depends on activation magnitudes, which are largely discarded in binary networks through the  $\text{sign}(\cdot)$  operation.
- **Spatial Information Loss:** Global averaging over spatial dimensions eliminates fine-grained structural patterns crucial for binary feature discrimination.
- **Gradient Incompatibility:** The covariance-based gradient flow conflicts with the straight-through estimator used in BNN training.

## 2. Spatial Approach (Our CC-Loss)

Our proposed Channel Correlation Loss operates directly on normalized spatial features, making it particularly suitable for binary networks.

**Formal Definition:**

$$\mathcal{L}_{CC} = \frac{1}{B} \sum_{b=1}^B \log \left( 1 + \frac{\beta}{\tau} \sum_{i=1}^C \sum_{j=i+1}^C \max(0, \tilde{\mathbf{z}}_i^{(b)T} \tilde{\mathbf{z}}_j^{(b)})^2 \right) \quad (2)$$

where  $\tilde{\mathbf{z}}_i^{(b)} = \frac{\mathbf{z}_i^{(b)}}{\|\mathbf{z}_i^{(b)}\|_2 + \epsilon}$  are L2-normalized feature vectors.

**Theoretical Foundations:**

**Angular Focus:** In binary networks where  $\text{sign}(\cdot)$  discards magnitude information, angular relationships become the primary carriers of discriminative information. Our cosine similarity measurement naturally aligns with this constraint.

**Asymmetric Competition:** By selectively penalizing only positive correlations:

$$\mathcal{L}_{CC} \propto \sum_{i < j} \max(0, \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_j)^2$$

we preserve potentially beneficial negative correlations that may encode complementary feature patterns, while actively discouraging redundant positive correlations.

**Progressive Regularization:** The logarithmic formulation with the temperature parameter  $\tau$  ensures smooth gradient transitions:

$$\frac{\partial \mathcal{L}_{CC}}{\partial z_i} \propto \frac{\max(0, \tilde{\mathbf{z}}_i^T \mathbf{z}_j)}{\tau + \beta \sum_{k \neq l} \max(0, \tilde{\mathbf{z}}_k^T \mathbf{z}_l)^2}$$

providing automatic gradient scaling and preventing early training disruption.

**Geometric Interpretation:** In the normalized feature space, CC-Loss promotes angular separation between channel vectors. For binary features, where magnitude information is discarded during inference, this angular diversity directly translates to enhanced discriminability in the Hamming space.

## 3. Information-Theoretic Approach (Mutual Information)

The information-theoretic approach aims to minimize mutual information between channels, promoting statistical independence in the information-theoretic sense.

**Formal Definition:**

$$\mathcal{L}_{\text{info}} = \sum_{i \neq j} I(\mathbf{z}_i; \mathbf{z}_j) \approx \sum_{i \neq j} \log \frac{\|\Sigma_{ij}\|_F^2}{\Sigma_{ii} \Sigma_{jj}} \quad (3)$$

where  $I(\mathbf{z}_i; \mathbf{z}_j)$  represents the mutual information between channels  $i$  and  $j$ , and the approximation leverages the relationship between mutual information and correlation for Gaussian distributions.

**Theoretical Basis:** This approach is grounded in the information bottleneck principle, which seeks to find representations that are maximally informative about the target while being maximally compressive about the input. By minimizing mutual information between channels:

$$\min \sum_{i \neq j} I(\mathbf{z}_i; \mathbf{z}_j)$$

we encourage the network to learn statistically independent feature extractors.

**Limitations in BNN Context:**

- **Continuous Distribution Assumption:** Mutual information estimators typically assume continuous distributions, while binary activations reside in discrete  $\{\pm 1\}$  spaces.

Table 1. Theoretical comparison of three decorrelation approaches

Method	Theoretical Basis	BNN Compatibility	Key Limitation
Statistical	Covariance minimization	Low	Magnitude sensitivity
Information-Theoretic	Mutual information	Medium	Discrete space mismatch
Spatial (Ours)	Angular diversity	High	Minimal overhead

- **Optimization Challenges:** Direct mutual information minimization in discrete spaces is notoriously difficult and often requires sophisticated estimators.
- **Computational Complexity:** Accurate mutual information estimation typically requires large batch sizes or memory-intensive techniques.

#### 4. Comparative Analysis

##### Key Insights:

- The **statistical approach** suffers from fundamental incompatibility with binary representations due to its reliance on magnitude information.
- The **information-theoretic approach**, while theoretically appealing, faces practical challenges in discrete optimization spaces.
- Our **spatial approach** successfully addresses these limitations by operating on angular relationships that align with the intrinsic properties of binary features.

The superior performance of our spatial approach, as demonstrated in our ablation studies, validates this theoretical analysis and establishes CC-Loss as the most suitable decorrelation method for Binary Neural Networks.

### Section C: Experimental Details

This section provides implementation details and layer selection analysis for CC-Loss, focusing on practical deployment considerations.

#### 1. Implementation Strategy

We adopted a simplified implementation approach that balances performance with practical deployment considerations. Rather than extracting features from specific intermediate layers within residual blocks, we apply CC-Loss to the output of complete layers, which provides a good trade-off between performance gains and implementation complexity.

**Feature Extraction:** For ResNet architectures, we extract features after the complete forward pass of layer2, which corresponds to intermediate-level features with sufficient complexity for structural optimization.

**Normalization:** Features are L2-normalized along spatial dimensions before computing cosine similarities, ensuring scale-invariant correlation measurements.

**Integration:** The CC-Loss is computed during training

and added to the primary classification loss with a progressive weighting scheme.

#### 2. Layer Selection Analysis

We evaluated CC-Loss application at different network depths to identify optimal placement strategies. Table 2 summarizes the performance variations across different layer positions.

Table 2. Layer-wise performance of CC-Loss on ResNet-20 (CIFAR-10, 1w1a)

Layer Position	Top-1 (%)	$\Delta$	Recommendation
Baseline	86.5	-	-
After Layer1	86.8	+0.3	Limited benefit
After Layer2	87.2	+0.7	<b>Recommended</b>
After Layer3	86.9	+0.4	Moderate benefit
After Layer4	86.4	-0.1	Not recommended

##### Early Layers (After Layer1)

Features at this stage represent basic patterns and show limited structural redundancy. While CC-Loss provides modest improvements (+0.3%), the benefits are constrained by the simplicity of early features.

##### Middle Layers (After Layer2)

This position represents the optimal balance, where features have developed sufficient complexity to benefit from structural optimization while not being over-specialized for final classification. We observe consistent improvements of +0.6-0.7% across different architectures.

##### Late Layers (After Layer3/Layer4)

Features in later layers become increasingly specialized for the classification task. Applying CC-Loss at these stages shows diminishing returns and can sometimes disrupt learned discriminative patterns.

### 3. Architecture Recommendations

Based on our experimental findings, we provide the following practical recommendations for CC-Loss deployment:

#### ResNet Family

For ResNet-18/20/34 architectures, apply CC-Loss after the second residual layer (layer2). This position consistently provides the best performance-computation trade-off.

## VGG Family

For VGG-Small, apply CC-Loss after the third convolutional block. The sequential nature of VGG architectures benefits from application after sufficient feature development.

## Implementation Simplicity

We recommend the simplified approach of applying CC-Loss to complete layer outputs rather than extracting features from specific intermediate positions within blocks. This approach provides 85-90% of the potential performance gains with significantly reduced implementation complexity.

## 4. Hyperparameter Settings

The effectiveness of CC-Loss depends on appropriate hyperparameter selection. Our experiments identified the following optimal ranges:

**Loss Weight ( $\beta$ ):**  $10^{-4}$  to  $10^{-3}$  provides the best balance between decorrelation and task performance. Values outside this range can lead to under-regularization or interference with primary task learning.

**Temperature ( $\tau$ ):** 0.1 ensures stable gradient computation and prevents numerical instability.

**Progressive Scheduling:** We employ a delayed activation strategy where CC-Loss weighting increases gradually after the first 50% of training epochs, allowing the network to establish stable representations before introducing structural constraints.

## 5. Computational Considerations

CC-Loss introduces minimal computational overhead during training:

- **Memory:** +2-3% due to feature storage for correlation computation
- **Time:** +2-4% per training epoch
- **Inference:** Zero overhead as CC-Loss is only applied during training

The spatial correlation computation scales quadratically with the number of channels but linearly with batch size, making it efficient for typical BNN deployment scenarios.