Sharpness-Aware Minimization Directly on the Boolean Hypercube

Anonymous Authors[1]

Abstract

Sharpness-Aware Minimization (SAM) improves generalization in continuous deep learning, yet applying it to binary-weight networks via latent-space heuristics suffers from a fundamental geometric mismatch: continuous perturbations do not faithfully probe the discrete loss landscape on $\{-1,+1\}^n$. We introduce BOLD-SAM, the first sharpness-aware optimizer that operates natively on the Boolean hypercube, replacing the Euclidean ℓ_p-ball with a k-bit Hamming ball and solving the resulting discrete min-max problem via greedy ascent followed by sharpness-aware bit-flip descent. The objective is theoretically justified from three complementary perspectives: PAC-Bayes, compression, and distributionally robust optimization. Experiments on a wide range of architectures and datasets demonstrate consistent improvements in clean accuracy and out-of-distribution robustness over standard binary training and latent-space SAM baselines.

1. Introduction

The landscape of deep learning is undergoing a structural transformation, driven by the tension between the computational demands of high-dimensional continuous optimization and the physical limitations of hardware at the edge. Traditional neural architectures rely on weight optimization within Euclidean space—a paradigm that has yielded immense success but at the cost of high energy consumption and substantial memory footprints. This has revitalized interest in discrete models, specifically Binarized Neural Networks (BNNs) [14, 32], e.g., BitNet [35], and the mathematically rigorous Boolean Logic Deep Learning (BOLD) framework [28, 34], which eliminates latent real-valued weights and operates directly on the Boolean hypercube $\{-1,+1\}^n$. However, optimizing over a discrete domain introduces fundamental challenges absent from continuous training: the loss landscape is defined on an exponentially large combinatorial space (2^n configurations), gradient-based methods cannot be applied directly, and finding a global minimum is NP-hard in general [3]. The loss surface exhibits complex interactions between bit-flips—synergistic effects where flipping two bits together produces a loss change far exceeding the sum of their individual contributions—making the landscape fundamentally different from the smooth manifolds of continuous optimization.

Despite the efficiency of these models, they inherit a fundamental vulnerability: the tendency of optimizers to converge toward *sharp* minima—weight configurations where minimal bit-flips cause catastrophic loss increases, degrading generalization. While the connection between flat minima and generalization is well-established in continuous deep learning [13, 16, 15], the relationship is subtle: [7] showed that common flatness measures are sensitive to reparameterization in continuous networks, complicating their use as generalization predictors. Sharpness-Aware Minimization (SAM) [10] has emerged as a principled remedy with strong theoretical foundations [21, 17], yet transferring these insights to the binary domain remains unexplored. Existing approaches apply SAM to BNNs through latent-weight proxies [22, 30], but these suffer from two geometric pathologies formalized in § 3: the perturbation signal *vanishes* as latent weights grow, and Euclidean proximity *fails to predict* Hamming proximity. Notably, the reparameterization ambiguities that plague continuous flatness measures [7] are *absent* on the Boolean hypercube: $\{-1,+1\}^n$ admits no smooth rescaling symmetries, so Hamming-ball sharpness is an unambiguous measure of the loss landscape geometry.

Contributions. This paper tackles these issues with BOLD-SAM, a native Boolean optimizer that replaces the continuous ℓ_p-ball with a k-bit Hamming ball—the natural neighborhood on $\{-1,+1\}^n$—and solves the resulting discrete min-max problem via a two-phase greedy procedure built on the Boolean Variation signal [28]. Our contributions include: ***(i)*** the objective is theoretically justified from three complementary perspectives—PAC-Bayes, compression, and distributionally robust optimization (§ 4); and ***(ii)*** experiments on CIFAR-10/100 with VGG, ResNet, and ViT demonstrate consistent improvements in both clean accuracy and out-of-distribution robustness over standard binary training and latent-space SAM baselines.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review for the ICML 2026 Workshop on Weight-Space Symmetries. Do not distribute.

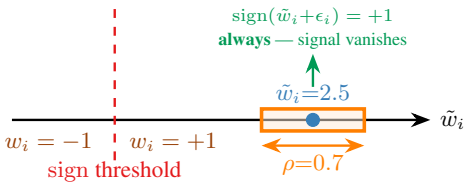

Figure 1. **The vanishing perturbation effect** (Proposition 3.1). Continuous SAM perturbs $\tilde{w}_i = 2.5$ within a ball of radius $\rho = 0.7$. Since the ball never crosses the sign threshold, the sharpness signal vanishes.

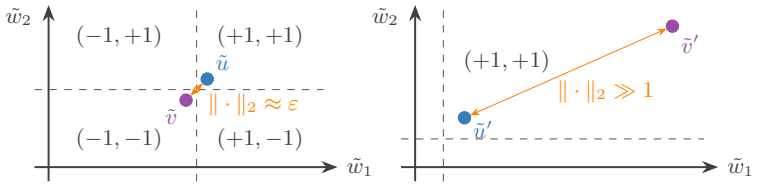

Figure 2. **Euclidean–Hamming metric distortion** (Proposition 3.2, n=2). *Left:* Two latent-weight vectors $\tilde{u}, \tilde{v}$ that are ε-close in $\mathbb{R}^2$ but straddle the sign boundary, producing $d_H = 2$. *Right:* Two vectors far apart but in the same quadrant, so $d_H = 0$. Euclidean proximity is neither necessary nor sufficient for Hamming proximity.

2. Related Work

SAM and its variants. [10] introduced SAM as a min-max optimizer seeking parameters with uniformly low loss in an ℓ_p-ball neighborhood. Subsequent work refined the perturbation geometry [21], reduced cost [23], and improved the ascent direction [18]. [17] recently established convergence of continuous SAM by viewing it as inexact gradient descent. All existing variants operate exclusively in continuous Euclidean parameter spaces.

Binary neural networks. Modern BNNs maintain latent real-valued weights and project them via the sign function, using the Straight-Through Estimator (STE) for gradient approximation [14, 32], with architectural advances [25, 24, 37, 36]. [22, 30] apply SAM to BNNs, but operate on latent continuous weights; [26] documented "perturbation diminishment" in quantized networks, partially anticipating our Vanishing Perturbation Effect. BOLD [28] eliminated latent weights, introducing Boolean Variation as a discrete gradient signal.

Flat minima in discrete spaces. [1] identified dense clusters of solutions in binary networks, and [2] formalized local entropy as a discrete flatness measure. Their optimization relies on replicated SGD and belief propagation; our objective (1) instead minimizes worst-case loss (min-max), offering different theoretical guarantees.

3. Pathology of Latent-Weight SAM

Applying continuous SAM to binarized networks via latent-weight proxies [26] introduces two geometric pathologies. Standard approaches perturb latent weights $\tilde{w} \in \mathbb{R}^n$ within an ℓ_p-ball of radius ρ before binarization $w = \text{sign}(\tilde{w})$. We formalize below why this indirect adversarial search inherently fails, necessitating a native Boolean SAM.

3.1. The Vanishing Perturbation Effect

Proposition 3.1 (Vanishing Perturbation). *Let $\mathcal{C}(\rho) = \{i : |\tilde{w}_i| > \rho\}$ be the "confident" coordinates. For any ϵ with $\|\epsilon\|_\infty \le \rho$, $\text{sign}(\tilde{w}_i + \epsilon_i) = \text{sign}(\tilde{w}_i)$ for all $i \in \mathcal{C}(\rho)$. If $|\mathcal{C}(\rho)| = n$, SAM reduces to standard gradient descent on the latent weights.*

The proof is available in Appendix C. The practical implication is severe: as training progresses and latent weights grow in magnitude, the fraction of confident coordinates $|\mathcal{C}(\rho)|/n$ approaches 1, and the continuous SAM objective degenerates to standard gradient descent (see Fig. 1 for an illustration).

3.2. Euclidean–Hamming Geometric Mismatch

The second failure stems from a geometric mismatch: flatness in the continuous latent space does not imply robustness in the actual discrete operating domain. Denote d_H as Hamming distance, we have:

Proposition 3.2 (Metric Distortion). *Map $\pi : \mathbb{R}^n \to \{-1,+1\}^n$ ($\pi(\tilde{w}) = \text{sign}(\tilde{w})$) can exhibit arbitrarily large metric distortion: for any $D > 0$, there exist $\tilde{u}, \tilde{v} \in \mathbb{R}^n$ with $\|\tilde{u} - \tilde{v}\|_2 < D^{-1}$ but $d_H(\pi(\tilde{u}), \pi(\tilde{v})) = n$, and conversely, $\tilde{u}', \tilde{v}'$ with $\|\tilde{u}' - \tilde{v}'\|_2 > D$ but $d_H(\pi(\tilde{u}'), \pi(\tilde{v}')) = 0$.*

The sign projection collapses continuous neighborhoods, inducing severe metric distortion: points straddling the origin can be arbitrarily ε-close in ℓ_2 yet maximally distant in Hamming space. Consequently, Euclidean proximity completely fails to capture discrete neighborhood structure (see Fig. 2 and Appendix C).

4. BOLD-SAM: Formulation and Algorithm

4.1. The Hamming-Ball Min-Max Objective

Let $w \in \{-1,+1\}^n$ denote the n Boolean weights in our model, $\mathcal{L}(w)$ the population loss, and $\hat{\mathcal{L}}(w) \equiv \hat{\mathcal{L}}(w; \mathcal{D})$ the empirical loss on training data $\mathcal{D}$ (we omit $\mathcal{D}$ when clear from context). Table 1 summarizes our main notations. The BOLD-SAM framework replaces the continuous ℓ_p-ball with the k-bit Hamming ball $\mathcal{B}_k(w) = \{w' \in \{-1,+1\}^n : d_H(w, w') \le k\}$ and solves:

$$\min_{w \in \{-1,+1\}^n} \mathcal{L}_{\text{adv}}(w) := \max_{w' \in \mathcal{B}_k(w)} \hat{\mathcal{L}}(w'; \mathcal{D}). \quad (1)$$

The adversarial loss is defined as the worst-case empirical loss over all configurations within k bit-flips of the current weights. This objective is not ad hoc—it is the native

Boolean analogue of the PAC-Bayes generalization bound that motivates the original continuous SAM objective [10]. We provide three independent justifications below. The detailed proofs are in Appendix D.

Justification I: PAC-Bayes upper bound (Boolean analogue of [10]). In continuous SAM, [10] use a Gaussian posterior $Q = \mathcal{N}(w, \rho^2 I)$ so that the expected loss is controlled by the worst case over the ℓ_2-ball. We replicate this argument on the Boolean hypercube using a Bernoulli bit-flip distribution. Consider this distribution P centered at w that independently flips each bit with probability $q = k/(2n)$. Splitting the expectation over $\mathcal{B}_k(w)$ and its complement, and bounding the tail mass of the Binomial distribution, $\beta_k = \Pr[\mathrm{Bin}(n, q) > k] \le e^{-k/6}$, via Chernoff [5, 4],:

$$\mathbb{E}_{w' \sim P}[\hat{\mathcal{L}}(w')] \;\le\; (1 - \beta_k)\, \mathcal{L}_{\mathrm{adv}}(w) + \beta_k B, \qquad (2)$$

where $B = \max_{w'} \hat{\mathcal{L}}(w')$. Combining with the PAC-Bayes–McAllester inequality [27]:

Theorem 4.1 (Discrete PAC-Bayes Sharpness Bound)**.** *For any prior P_0 independent of the training set $\mathcal{S}$ of size m, and any $\delta \in (0, 1)$, with probability $\ge 1 - \delta$:*

$$\mathbb{E}_{w' \sim P}\big[\mathcal{L}(w')\big] \;\le\; (1 - \beta_k)\, \mathcal{L}_{\mathrm{adv}}(w) + \beta_k B + \sqrt{\frac{\mathrm{KL}(P \| P_0) + \log(2\sqrt{m}/\delta)}{2m}}, \qquad (3)$$

where $P = \mathrm{BernoulliFlip}(w, q)$ with $q = k/(2n)$, $\beta_k \le e^{-k/6}$, and $B = \max_{w'} \hat{\mathcal{L}}(w')$.

The structure mirrors [10]: the sharpness-aware term replaces $\max_{\|\epsilon\| \le \rho} \hat{\mathcal{L}}(w+\epsilon)$, with $\beta_k \le e^{-k/6}$ replacing the Gaussian tail and Bernoulli KL replacing Gaussian KL. The crucial difference is that our bound operates natively on $\{-1, +1\}^n$. Since β_k is exponentially small for practical k, the bound is effectively $\mathbb{E}_{w' \sim P}[\mathcal{L}(w')] \le \mathcal{L}_{\mathrm{adv}}(w) + \mathrm{KL}$ term, directly motivating the minimization of $\mathcal{L}_{\mathrm{adv}}(w)$. Full proof in Appendix D.

Justification II: Compression bound. If the BOLD-SAM solution $\hat{w}$ satisfies $\mathcal{L}_{\mathrm{adv}}(\hat{w}) \le \epsilon$, *every* configuration in $\mathcal{B}_k(\hat{w})$ also has empirical loss $\le \epsilon$. Because specifying just $\hat{w}$ describes this entire set, the effective hypothesis class reduces from 2^n to $\le 2^n / V_k$ via a Hamming-ball covering argument:

$$\mathcal{L}(\hat{w}) \le \epsilon + \sqrt{\frac{n \log 2 - \log V_k + \log(1/\delta)}{2m}}, \qquad (4)$$

where $V_k = \sum_{j=0}^{k} \binom{n}{j}$. The term $\log V_k \approx k \log(n/k)$ quantifies the complexity reduction from Hamming-ball flatness: for $k = \Theta(\sqrt{n})$, this scales as $\Theta(\sqrt{n} \log n)$, providing substantial reduction over the raw $n \log 2$ complexity. Full derivation in Appendix D.2.

Justification III: Distributionally robust optimization. The objective (1) admits a natural interpretation as DRO [31]: $\mathcal{L}_{\mathrm{adv}}(w) = \max_{\mu \in \Delta(\mathcal{B}_k(w))} \mathbb{E}_{w' \sim \mu}[\hat{\mathcal{L}}(w')]$, where $\Delta(\mathcal{B}_k(w))$ is the set of all distributions supported on the Hamming ball. Because the maximizing distribution is the Dirac measure at the worst-case point, BOLD-SAM guards against the worst-case realization within the Hamming ball, the natural discrete counterpart of a Wasserstein ambiguity set [20] (Appendix D.3).

Remark 4.2 (The role of k). All three derivations reveal k as a bias-sharpness tradeoff: *(i)* in the PAC-Bayes view, larger k makes β_k smaller (better concentration) but forces $\mathcal{L}_{\mathrm{adv}}$ higher; *(ii)* in the compression view, larger k increases $\log V_k$ (more complexity reduction) but requires uniformly low loss over a larger set; *(iii)* in the DRO view, larger k gives more conservative population coverage but a harder optimization problem.

4.2. Boolean Variation as a Discrete Gradient

The methodology employs the Boolean Variation framework of [28], which introduced a calculus of discrete derivatives for Boolean functions equipped with a chain rule that enables backpropagation through deep Boolean networks. The reader can check Appendix B for a brief review. The BOLD variation of the loss with respect to weight w_i is defined as:

$$f'_i(w) \;=\; \mathrm{xnor}\big(\delta(w_i \to \neg w_i),\; \delta\hat{\mathcal{L}}(w \to w^{(i)})\big), \qquad (5)$$

where $\delta(\cdot \to \cdot)$ denotes the Boolean variation (direction of change) and $w^{(i)}$ is w with bit i negated. In the standard $\{-1, +1\}$ conversion (where xnor corresponds to multiplication), this evaluates to $f'_i(w) = -w_i\,[\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)]$. The sign convention encodes the descent direction: a bit is flipped when $f'_i \cdot w_i > 0$ (equivalently, $\mathrm{BV}_i < 0$), meaning the flip *decreases* the loss, mirroring the role of the negative gradient in continuous optimization. BOLD's chain rule computes f'_i for all n weights in a single backward pass.

For the theoretical analysis in this paper, it is more natural to work with the *unsigned loss change*:

$$\mathrm{BV}_i(w) \;=\; \hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w) \;=\; -w_i \cdot f'_i(w), \qquad (6)$$

which is the standard first-order discrete derivative of pseudo-Boolean function theory [6, 29]. The two conventions are interchangeable and we can recover one from the other. We use BV_i throughout the theoretical sections for cleaner notation, and well-aligned with the standard literature on discrete analysis. In the algorithm, Phase 2 reduces to BOLD's standard flip rule (Eq. 9 of [28]) when expressed in the f'_i convention: a bit is flipped when $\mathrm{xnor}(f'_i, w_i) = \mathrm{T}$, equivalently when $\mathrm{BV}_i < 0$.

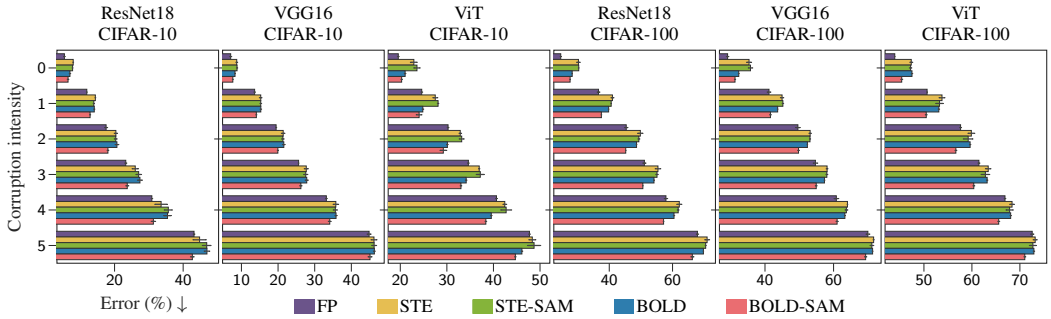

Figure 3. Error rate (%, ↓) at corruption intensities for different architectures on CIFAR-10 and CIFAR-100.

4.3. The BOLD-SAM Algorithm

Algorithm 1 illustrates our optimization procedure. **Phase 1** greedily identifies the k bits that most increase the loss, constructing an adversarial point in $\mathcal{B}_k(w^t)$. **Phase 2** evaluates the Boolean Variation at this *worst-case* point and flips the single bit that most reduces the loss—updating the original weights, not the adversarial ones. Computing descent at this worst-case point ensures robustness against sharp loss landscapes. Using BOLD's chain rule (Appendix B), each "Compute BV_i" loop requires just one backward pass. The total per-step cost is therefore two backward passes (at w^t and $\hat{w}$), comparable to continuous SAM.

Algorithm 1: BOLD-SAM Optimization Step

Input: Current weights $w^t \in \{-1, +1\}^n$, Hamming radius k

Phase 1: Greedy Ascent (Adversarial Probing)
for *each bit* $i = 1, \ldots, n$ **do**
  Compute $\mathrm{BV}_i(w^t)$;
// k indices with largest BV
$\mathcal{I} \leftarrow \text{argtop-k}\{\mathrm{BV}_i(w^t)\}_{i=1}^n$
// Adversarial point $\hat{w} \in \mathcal{B}_k(w^t)$
$\hat{w} \leftarrow w^t$ with bits in $\mathcal{I}$ flipped
Phase 2: Sharpness-Aware Descent (Weight Update)
for *each bit* $i = 1, \ldots, n$ **do**
  Compute $\mathrm{BV}_i(\hat{w})$;
// Exclude ascent bits
$i^* \leftarrow \arg\min_{i \notin \mathcal{I}} \mathrm{BV}_i(\hat{w})$
if $\mathrm{BV}_{i^*}(\hat{w}) < 0$ **then**
  $w^{t+1} \leftarrow w^t$ with bit i^* flipped;
else
  $w^{t+1} \leftarrow w^t$;

5. Experiments

We evaluate BOLD-SAM on ResNet-18 [11], VGG-16 [33], and ViT [8] using CIFAR-10/100 [19] and their corrupted variants [12]. All binary methods use identical architectures with standard weights replaced by Boolean (BOLD) or binary (STE) parameters. We compare: full-precision (FP), standard STE [14], latent-space STE-SAM [10], standard BOLD, and BOLD-SAM (k=500). Fig. 3 reports error rates at corruption intensities 0 (clean) through 5 across all architecture–dataset combinations.

BOLD-SAM improves over all Boolean baselines. On clean CIFAR-10, BOLD-SAM reduces the error of standard BOLD across architectures. On CIFAR-100 the gains are comparable. STE-SAM, which applies continuous SAM to latent weights, provides smaller and less consistent improvements over STE—confirming that the vanishing perturbation effect (§ 3) limits the efficacy of latent-space sharpness-aware training.

Robustness gains exceed clean accuracy gains. As corruption intensity increases, the gap between BOLD-SAM and BOLD widens. The robustness improvement is typically 2–3× the clean accuracy improvement, consistent with the theoretical prediction that flat minima (low B_k) transfer better under distribution shift. On several corruption types, BOLD-SAM narrows the gap to the full-precision baseline substantially, despite using only 1-bit weights—suggesting that the discrete flatness enforced by Hamming-ball perturbations provides an effective form of robustness complementary to the representational capacity of full-precision models.

6. Conclusion

We introduced BOLD-SAM, the first sharpness-aware optimizer that operates natively on the Boolean hypercube, eliminating the geometric pathologies inherent in latent-space approaches. By systematically probing the discrete loss landscape within a k-bit Hamming ball, our experiments demonstrate that BOLD-SAM yields consistent improvements in clean accuracy and significant gains in out-of-distribution robustness. We believe the native discrete optimization framework developed here—particularly the use of Boolean Variation for direct adversarial probing—offers a principled and highly effective foundation for future research in robust, energy-efficient binary neural networks.

References

[1] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina. Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses. *Physical Review Letters*, 115(12):128101, 2015.

[2] C. Baldassi, F. Pittorino, and R. Zecchina. Shaping the Learning Landscape in Neural Networks around Wide Flat Minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020.

[3] A. Blum and R. Rivest. Training a 3-node Neural Network is NP-complete. *Advances in Neural Information Processing Systems (NIPS)*, 1, 1988.

[4] S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. ISBN 9780199535255. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199535255.001.0001.

[5] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[6] Y. Crama and P. L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2011.

[7] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp Minima Can Generalize For Deep Nets. In *International Conference on Machine Learning (ICML)*, 2017.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[9] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, New York, 3rd edition, 1968. ISBN 978-0-471-25708-0.

[10] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[13] S. Hochreiter and J. Schmidhuber. Flat Minima. *Neural Computation*, 9(1):1–42, 1997.

[14] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[15] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic Generalization Measures and Where to Find Them. In *International Conference on Learning Representations (ICLR)*, 2020.

[16] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *International Conference on Learning Representations (ICLR)*, 2017.

[17] P. D. Khanh, H.-C. Luong, B. S. Mordukhovich, and D. B. Tran. Fundamental Convergence Analysis of Sharpness-Aware Minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[18] M. Kim, D. Li, S. X. Hu, and T. Hospedales. Fisher SAM: Information Geometry and Sharpness Aware Minimisation. In *International Conference on Machine Learning (ICML)*, 2022.

[19] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009.

[20] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh. Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning. *Operations Research & Management Science in the Age of Analytics*, pages 130–166, 2019.

[21] J. Kwon, J. Kim, H. Park, and I. K. Choi. ASAM: Adaptive Sharpness-Aware Minimization for Scale-Invariant Learning of Deep Neural Networks. In *International Conference on Machine Learning (ICML)*, 2021.

[22] R. Liu, F. Bian, and X. Zhang. Binary Quantized Network Training with Sharpness-Aware Minimization. *Journal of Scientific Computing*, 94(1):16, 2023.

[23] Y. Liu, S. Mai, X. Chen, C.-J. Hsieh, and Y. You. Towards Efficient and Scalable Sharpness-Aware Minimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng. Bi-Real Net: Enhancing the Performance of 1-bit CNNs With Improved Representational Capability and Advanced Training Algorithm. In *European Conference on Computer Vision (ECCV)*, 2018.

[25] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng. ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions. In *European Conference on Computer Vision (ECCV)*, 2020.

[26] Z. Liu, K.-T. Cheng, D. Huang, E. P. Xing, and Z. Shen. Nonuniform-to-Uniform Quantization: Towards Accurate Quantization via Generalized Straight-Through Estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. Sharpness-aware quantization (SAQ) component.

[27] D. A. McAllester. PAC-Bayesian Stochastic Model Selection. *Machine Learning*, 51:5–21, 2003.

[28] V. M. Nguyen, C. Ocampo, A. Askri, L. Leconte, and B.-H. Tran. BOLD: Boolean Logic Deep Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[29] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[30] H. Pu, D. Zhang, K. Xu, R. Mo, Z. Yan, and D. Wang. BNN-SAM: Improving Generalization of Binary Object Detector by Seeking Flat Minima. *Applied intelligence*, 54(8):6682–6700, 2024.

[31] H. Rahimian and S. Mehrotra. Frameworks and Results in Distributionally Robust Optimization. *Open Journal of Mathematical Optimization*, 3:1–85, 2022.

[32] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *European Conference on Computer Vision (ECCV)*, 2016.

[33] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[34] B.-H. Tran and V. M. Nguyen. Highly Efficient and Effective LLMs with Multi-Boolean Architectures. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=r0CH5dF3Se.

[35] H. Wang, S. Ma, L. Ma, L. Wang, W. Wang, L. Dong, S. Huang, H. Wang, J. Xue, R. Wang, Y. Wu, and F. Wei. BitNet: 1-bit Pre-training for Large Language Models. *Journal of Machine Learning Research*, 26(125):1–29, 2025. URL http://jmlr.org/papers/v26/24-2050.html.

[36] X. Xing, Y. Li, W. Li, W. Ding, Y. Jiang, Y. Wang, J. Shao, C. Liu, and X. Liu. Towards accurate binary neural networks via modeling contextual dependencies. In *European Conference on Computer Vision (ECCV)*, 2022.

[37] Y. Zhang, Z. Zhang, and L. Lew. PokeBNN: A Binary Pursuit of Lightweight Accuracy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

A. Notations

The main notation used throughout the paper is summarized in Table 1.

Table 1. Summary of main notations.

Symbol	Description
Setup	
n	Number of Boolean parameters
m	Number of training samples ($\lvert\mathcal{D}\rvert$ or $\lvert\mathcal{S}_1\rvert$)
$\mathcal{D}$	Training dataset
$w \in \{-1,+1\}^n$	Boolean weight vector
$w^{(i)}$	w with bit i negated
$w^{(ij)}$	w with bits i and j negated
$w^t \oplus \mathbf{1}_S$	w^t with all bits in set S flipped
Loss functions	
$\mathcal{L}(w)$	Population (true) loss: $\mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(f_w(x),y)]$
$\hat{\mathcal{L}}(w)$	Empirical loss: $\frac{1}{m}\sum_{i=1}^m \ell(f_w(x_i),y_i)$
$\mathcal{L}_{\text{adv}}(w)$	Adversarial loss: $\max_{w'\in\mathcal{B}_k(w)} \hat{\mathcal{L}}(w')$
B	Global loss bound: $\max_{w'} \hat{\mathcal{L}}(w')$
B_k	Sharpness: $\mathcal{L}_{\text{adv}}(w) - \hat{\mathcal{L}}(w)$
Hamming geometry	
$d_H(w,w')$	Hamming distance between w and w'
$\mathcal{B}_k(w)$	Hamming ball: $\{w' : d_H(w,w') \le k\}$
V_k	Hamming ball volume: $\sum_{j=0}^k \binom{n}{j}$
k	Hamming radius (adversarial probe strength)
Discrete calculus	
$f'_i(w)$	BOLD variation (signed): $-w_i[\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)]$
$\text{BV}_i(w)$	Boolean Variation (unsigned): $\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)$
$H_{ij}(w)$	Interaction Hessian: $\hat{\mathcal{L}}(w^{(ij)}) - \hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w^{(j)}) + \hat{\mathcal{L}}(w)$
h	Hessian bound: $\max_{w,\, i\neq j} \lvert H_{ij}(w)\rvert$
PAC-Bayes	
$\mathcal{S}_0, \mathcal{S}_1$	Data split for prior / posterior training
w_0	Prior center (trained on $\mathcal{S}_0$)
d	Adaptation distance: $d_H(\hat{w}, w_0)$
p	Bernoulli flip probability in the prior
β_k	Tail mass outside $\mathcal{B}_k(w)$ ($\le e^{-k/6}$)
δ	Confidence parameter (bound holds w.p. $\ge 1-\delta$)

B. Preliminaries: The BOLD Framework

We briefly review the Boolean Logic Deep Learning (BOLD) framework [28], which provides the foundation for BOLD-SAM. BOLD eliminates real-valued latent weights entirely: network parameters live natively in $\{-1,+1\}^n$ (equivalently $\{\text{T},\text{F}\}^n$ in Boolean logic).

B.1. Boolean Neuron

A BOLD neuron with input size m, weights $w_0, w_1, \ldots, w_m \in \{-1,+1\}$, and Boolean or real inputs computes the pre-activation:

$$s = w_0 + \textstyle\sum_{i=1}^m \text{L}(w_i, x_i), \tag{7}$$

where L is a Boolean logic gate (typically XNOR) and the summation counts the number of TRUE outputs. The framework is flexible, as it allows Boolean linear layers to be connected thorugh activation layers, normalization layers, or other types of layers.

B.2. Boolean Variation and Chain Rule

The key theoretical contribution of BOLD is a discrete derivative—the *Boolean Variation*—equipped with a chain rule that enables backpropagation. For a function $\hat{\mathcal{L}}$ and weight $w_i \in \{-1, +1\}$, the BOLD variation is:

$$f'_i(w) \;=\; \mathrm{xnor}\big(\delta(w_i \to \neg w_i),\; \delta\hat{\mathcal{L}}(w \to w^{(i)})\big), \tag{8}$$

where $\delta(\cdot \to \cdot)$ denotes the direction of change and $w^{(i)}$ is w with bit i negated. Intuitively, $f'_i = \mathrm{T}$ when $\hat{\mathcal{L}}$ co-varies with w_i. In the $\{-1, +1\}$ embedding (where xnor corresponds to multiplication; Proposition A.2 of [28]):

$$f'_i(w) = -w_i \left[\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)\right]. \tag{9}$$

The sign convention encodes the descent direction: $f'_i > 0$ when flipping bit i *decreases* the loss, mirroring the role of the negative gradient.

The critical property is the *Boolean chain rule* (Theorem 3.12 of [28]): for compositions $\mathbb{B} \xrightarrow{f} \mathbb{Z} \xrightarrow{g} \mathbb{R}$, the variation satisfies $(g \circ f)'(x) = \mathrm{xnor}(g'(f(x)), f'(x))$ under some assumptions. This enables computation of f'_i for all n weights in a single backward pass through a deep network.

B.3. Accumulator-Based Optimizer

The per-sample atomic variation $q_{i,k} = \mathrm{xnor}(\delta\mathcal{L}/\delta x_k^{l+1}, x_k^l)$ (Eq. 5 of [28]) is aggregated over the batch into an integer signal q_i (Eq. 7 of [28]). The simplest update rule flips w_i when $\mathrm{xnor}(q_i, w_i) = \mathrm{T}$—i.e., when the aggregated signal indicates that flipping decreases the loss.

In practice, BOLD uses an *accumulator* m_i^t that integrates signals across iterations:

$$m_i^{t+1} = \beta^t m_i^t + \eta^t q_i^{t+1}, \tag{10}$$

where η^t is a learning rate and $\beta^t = N^t_{\mathrm{unchanged}}/N_{\mathrm{total}}$ is a *plasticity ratio* that down-weights the accumulator when many bits are flipping (high plasticity) and preserves it when the network is stable (low plasticity). A bit is flipped when $\mathrm{xnor}(m_i, w_i) = \mathrm{T}$ (equivalently, $m_i \cdot w_i \geq 1$ in the embedding), and the accumulator resets to zero on flip. This provides an implicit, adaptive convergence threshold.

B.4. Notation Bridge: Unsigned Loss Change

For the theoretical analysis in this paper, it is more natural to work with the *unsigned loss change*:

$$\mathrm{BV}_i(w) \;=\; \hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w), \tag{11}$$

which is the standard first-order discrete derivative of pseudo-Boolean function theory [6, 29].

According to BOLD, the variation of the function f with respect to the i-th element of the parameters w is defined as $f'_i(w) \;=\; \mathrm{xnor}\big(\delta(w_i \to \neg w_i), \delta\hat{\mathcal{L}}(w_i \to \neg w_i)\big)$. We have $\delta\hat{\mathcal{L}}(w_i \to \neg w_i) = \delta\hat{\mathcal{L}}(w \to w^{(i)})$, as defined in [28]. Since $\hat{\mathcal{L}} : \{-1, +1\}^n \to \mathbb{R}$ has a numeric codomain, the variation $\delta\hat{\mathcal{L}}(w \to w^{(i)})$ is the standard finite difference $\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)$. Using the conversion $e(\mathrm{T}) = +1, e(\mathrm{F}) = -1$, and Proposition A.3(1) ($\mathrm{xnor}(a, x) = e(a) \cdot x$, for $a \in \mathbb{B}$ and $x \in \mathbb{R}$), we obtain

$$f'_i(w) = e(\delta(w_i \to \neg w_i)) \cdot \left[\hat{\mathcal{L}}(w^{(i)}) - \hat{\mathcal{L}}(w)\right]. \tag{12}$$

Now $e(\delta(w_i \to \neg w_i)) = -w_i$; if $w_i = +1$, then $\neg w_i = -1$, the direction is "down", $\delta = \mathrm{F}, e = -1 = -w_i$; if $w_i = -1$, then $\neg w_i = +1$, direction is "up", $\delta = \mathrm{T}, e = +1 = -w_i$. Therefore, we have:

$$f'_i(w) = -w_i \cdot \mathrm{BV}_i(w). \tag{13}$$

This means BOLD's variation is our BV multiplied by a known, bit-dependent sign. Positive f'_i means "flipping this bit decreases the loss"–gradient-like convention. Positive BV means "flipping this bit increase the loss"–raw loss-change convention. Since w_i^2, the two conventions are interchangeable and we can always recover one from the other, i.e., $\mathrm{BV}_i = -w_i \cdot f'_i$.

We use BV_i throughout the theoretical sections because it eliminates sign factors in the set function $f(S) = \hat{\mathcal{L}}(w^t \oplus \mathbf{1}_S) - \hat{\mathcal{L}}(w^t)$, the discrete Hessian, and the convergence analysis. In Algorithm 1, Phase 2 reduces to BOLD's standard flip rule when expressed in the f'_i convention: a bit is flipped when $\mathrm{xnor}(f'_i, w_i) = \mathrm{T}$, equivalently when $\mathrm{BV}_i < 0$.

C. Proofs for Pathological Behavior of Latent-Weight SAM

The primary motivation for developing a native Boolean SAM arises from the theoretical contradictions inherent in training binarized networks via latent-weight approximations. In the prevailing paradigm, high-precision latent weights $\tilde{w} \in \mathbb{R}^n$ are maintained and projected onto the Boolean space through a non-differentiable sign function, $w = \operatorname{sign}(\tilde{w}) \in \{-1, +1\}^n$ [14]. Gradients are approximated using the STE. When continuous SAM is applied to these latent weights, it searches for an adversarial perturbation ϵ within an ℓ_p-norm ball of radius ρ to maximize the loss:

$$\max_{\|\epsilon\|_p \le \rho} \mathcal{L}(\tilde{w} + \epsilon). \tag{14}$$

This approach suffers from two critical failures.

C.1. The Vanishing Perturbation Effect

The sign function $\operatorname{sign}(\cdot)$ acts as a quantization barrier. For a latent weight $\tilde{w}_i$ that has been optimized to high magnitude—representing high confidence—a perturbation ϵ bounded by ρ is frequently insufficient to cross the zero threshold. Consequently, the binarized network's functional output remains identical: $\operatorname{sign}(\tilde{w}_i + \epsilon_i) = \operatorname{sign}(\tilde{w}_i)$ for all $|\epsilon_i| \le \rho < |\tilde{w}_i|$. The adversarial loss equals the clean loss, and the sharpness-aware signal vanishes—the optimizer effectively reverts to standard SGD. This phenomenon is a binary-specific instance of the perturbation diminishment documented by [26] for multi-bit quantized networks.

Proposition C.1 (Vanishing Perturbation)**.** *Let $\tilde{w} \in \mathbb{R}^n$ be latent weights and $w = \operatorname{sign}(\tilde{w})$. Define $\mathcal{C}(\rho) = \{i \in [n] : |\tilde{w}_i| > \rho\}$ as the set of "confident" coordinates. Then for any ϵ with $\|\epsilon\|_\infty \le \rho$, the binarized adversarial point satisfies $\operatorname{sign}(\tilde{w}_i + \epsilon_i) = \operatorname{sign}(\tilde{w}_i)$ for all $i \in \mathcal{C}(\rho)$. In particular, if $|\mathcal{C}(\rho)| = n$ (all weights confident), then $\max_{\|\epsilon\|_\infty \le \rho} \mathcal{L}(\operatorname{sign}(\tilde{w} + \epsilon)) = \mathcal{L}(\operatorname{sign}(\tilde{w}))$, and SAM reduces to standard gradient descent on the latent loss.*

Proof. If $|\tilde{w}_i| > \rho \ge |\epsilon_i|$, then $\tilde{w}_i + \epsilon_i$ has the same sign as $\tilde{w}_i$. The second claim follows by noting that the binarized output is identical for all perturbations in the ℓ_∞-ball. □

C.2. Euclidean–Hamming Geometric Mismatch

The second failure stems from a geometric mismatch: flatness in the continuous latent space does not imply robustness in the actual discrete operating domain. Two latent configurations that are proximate in Euclidean distance may map to Boolean configurations that are far apart in Hamming distance, and vice versa.

Proposition C.2 (Metric Distortion)**.** *The map $\pi : \mathbb{R}^n \to \{-1, +1\}^n$ defined by $\pi(\tilde{w}) = \operatorname{sign}(\tilde{w})$ can exhibit arbitrarily large metric distortion: for any $D > 0$, there exist $\tilde{u}, \tilde{v} \in \mathbb{R}^n$ with $\|\tilde{u} - \tilde{v}\|_2 < D^{-1}$ but $d_H(\pi(\tilde{u}), \pi(\tilde{v})) = n$, and conversely, $\tilde{u}', \tilde{v}'$ with $\|\tilde{u}' - \tilde{v}'\|_2 > D$ but $d_H(\pi(\tilde{u}'), \pi(\tilde{v}')) = 0$.*

Proof. For the first claim, take $\tilde{u}_i = 1/(2nD)$ and $\tilde{v}_i = -1/(2nD)$ for all i; then $\|\tilde{u} - \tilde{v}\|_2 = 1/(D\sqrt{n}) < D^{-1}$ but $d_H(\pi(\tilde{u}), \pi(\tilde{v})) = n$. For the second, take $\tilde{u}'_i = 1$ and $\tilde{v}'_i = 1 + 2D/\sqrt{n}$; then $\|\tilde{u}' - \tilde{v}'\|_2 = 2D > D$ but $\pi(\tilde{u}') = \pi(\tilde{v}')$. □

D. Justifications for the Objective

Structural parallel with [10]. The foundational theorem of continuous SAM [10] establishes that, for any $\rho > 0$, with high probability over $S \sim \mathcal{D}^n$:

$$\mathcal{L}_{\mathcal{D}}(w) \le \underbrace{\max_{\|\epsilon\|_2 \le \rho} \hat{\mathcal{L}}_S(w + \epsilon)}_{\text{sharpness-aware loss}} + h\big(\|w\|_2^2/\rho^2\big), \tag{15}$$

where h is a strictly increasing function incorporating the KL divergence between a Gaussian posterior $Q = \mathcal{N}(w, \rho^2 I)$ and a grid of Gaussian priors. The continuous SAM objective $\min_w \max_{\|\epsilon\|_2 \le \rho} \hat{\mathcal{L}}_S(w + \epsilon)$ is then motivated directly by minimizing the right-hand side of (15).

BOLD-SAM's Eq. (1) replicates this derivation structure on the Boolean hypercube, with three systematic replacements:

Component	Foret et al. (continuous)	BOLD-SAM (Boolean)
Parameter space	$\mathbb{R}^d$	$\{-1,+1\}^n$
Perturbation set	ℓ_2-ball $\{\epsilon : \|\epsilon\|_2 \le \rho\}$	Hamming ball $\mathcal{B}_k(w)$
Stochastic posterior	$Q = \mathcal{N}(w, \rho^2 I)$	$Q = \mathrm{BernoulliFlip}(w, q)$
Complexity term	$h(\|w\|_2^2/\rho^2)$ via Gaussian KL	$d_H(\hat{w}, w_0)\cdot\log(1/p)$ via Bernoulli KL

We now make this analogy precise. We provide three independent theoretical justifications: a PAC-Bayes upper bound (the direct Boolean counterpart of [10]), a compression-based argument, and a distributionally robust optimization perspective.

D.1. Derivation I: PAC-Bayes Upper Bound

In continuous SAM, [10] use a Gaussian posterior $Q = \mathcal{N}(w, \rho^2 I)$ so that $\mathbb{E}_{\epsilon\sim Q}[\hat{\mathcal{L}}(w+\epsilon)]$ is controlled by $\max_{\|\epsilon\|_2\le\rho} \hat{\mathcal{L}}(w+\epsilon)$ via tail concentration of the χ^2 distribution. We replicate this argument on the Boolean hypercube using a Bernoulli bit-flip distribution as the stochastic posterior.

Consider the bit-flip distribution P centered at w that independently flips each bit with probability $q \in (0, \frac{1}{2})$. For any $w' \in \{-1,+1\}^n$ with $d_H(w', w) = j$, the distribution assigns mass $P(w') = q^j(1-q)^{n-j}$. The expected empirical loss under P is:

$$\mathbb{E}_{w'\sim P}\big[\hat{\mathcal{L}}(w')\big] = \sum_{j=0}^{n} \sum_{w' : d_H(w',w)=j} q^j(1-q)^{n-j}\, \hat{\mathcal{L}}(w').$$

Proposition D.1 (Adversarial Loss as a Stochastic Upper Bound)**.** *For any $w \in \{-1,+1\}^n$ and any $q \in (0, \frac{1}{2})$:*

$$\mathbb{E}_{w'\sim P}\big[\hat{\mathcal{L}}(w')\big] \;\le\; \underbrace{\Big[1 - \sum_{j=0}^{k} \tbinom{n}{j} q^j(1-q)^{n-j}\Big]}_{\triangleq\ \beta_k} \cdot B \;+\; \big(1-\beta_k\big)\cdot \mathcal{L}_{\mathrm{adv}}(w), \tag{16}$$

where $B = \max_{w'} \hat{\mathcal{L}}(w')$ is the global loss bound and β_k is the probability mass outside $\mathcal{B}_k(w)$. In particular, for $q \le k/(2n)$, we have $\beta_k \le e^{-k/6}$ (by a Chernoff bound), so the expected loss is dominated by $\mathcal{L}_{\mathrm{adv}}(w)$.

Proof. Split the expectation over the Hamming ball and its complement:

$$\mathbb{E}_{w'\sim P}[\hat{\mathcal{L}}(w')] = \underbrace{\sum_{\substack{w':\\ d_H(w',w)\le k}} P(w')\,\hat{\mathcal{L}}(w')}_{\le\ (1-\beta_k)\cdot\mathcal{L}_{\mathrm{adv}}(w)} + \underbrace{\sum_{\substack{w':\\ d_H(w',w)> k}} P(w')\,\hat{\mathcal{L}}(w')}_{\le\ \beta_k\cdot B}. \tag{17}$$

The first inequality holds because $\hat{\mathcal{L}}(w') \le \mathcal{L}_{\mathrm{adv}}(w)$ for all $w' \in \mathcal{B}_k(w)$ and the total mass on $\mathcal{B}_k(w)$ is $1-\beta_k$. The second holds since $\hat{\mathcal{L}}(w') \le B$ everywhere.

For the tail bound, the number of bit-flips under P is $\mathrm{Bin}(n, q)$ with mean $\mu = nq = k/2$ when $q = k/(2n)$. The standard multiplicative Chernoff bound [4] at $\delta = 1$ (i.e., $\Pr[X > 2\mu]$) gives:

$$\beta_k = \Pr[\mathrm{Bin}(n,q) > k] \le \Big(\frac{e}{4}\Big)^{\mu} = \Big(\frac{e}{4}\Big)^{k/2} = e^{-\frac{\ln 4 - 1}{2}\,k} \le e^{-k/6},$$

□

Combining this with the standard PAC-Bayes theorem yields the Boolean analogue of [10]'s generalization bound:

Theorem 4.1 (Discrete PAC-Bayes Sharpness Bound—Boolean Analogue of [10]). *Let P_0 be a prior over $\{-1,+1\}^n$ independent of the training set $\mathcal{S}$ of size m, and let P be the Bernoulli bit-flip distribution centered at w with flip probability $q = k/(2n)$. For any $\delta \in (0,1)$, with probability at least $1-\delta$ over $\mathcal{S} \sim \mathcal{D}^m$:*

$$\mathbb{E}_{w'\sim P}\big[\mathcal{L}(w')\big] \;\le\; \underbrace{(1-\beta_k)\cdot \mathcal{L}_{\mathrm{adv}}(w) + \beta_k B}_{\textit{sharpness-aware empirical term}} \;+\; \underbrace{\sqrt{\frac{\mathrm{KL}(P\|P_0) + \log(2\sqrt{m}/\delta)}{2m}}}_{\textit{complexity term}}, \tag{18}$$

where $\beta_k \le e^{-k/6}$ and the KL term depends on the choice of prior P_0.

Proof. By the PAC-Bayes–McAllester inequality [27], for any posterior Q and data-independent prior P_0:

$$\mathbb{E}_{w'\sim Q}[\mathcal{L}(w')] \;\le\; \mathbb{E}_{w'\sim Q}[\hat{\mathcal{L}}(w')] \;+\; \sqrt{\frac{\mathrm{KL}(Q\|P_0) + \log(2\sqrt{m}/\delta)}{2m}}.$$

Set $Q = P$ (the Bernoulli bit-flip distribution centered at w). By Proposition D.1, the expected empirical loss satisfies $\mathbb{E}_{w'\sim P}[\hat{\mathcal{L}}(w')] \le (1-\beta_k)\,\mathcal{L}_{\mathrm{adv}}(w) + \beta_k B$. Substituting yields (18). □

Remark D.3 (One-to-one correspondence with [10]). The structure of Theorem 4.1 mirrors [10] exactly: *(i)* the sharpness-aware term $(1-\beta_k)\mathcal{L}_{\mathrm{adv}}(w) + \beta_k B$ plays the role of $\max_{\|\epsilon\|_2\le\rho} \hat{\mathcal{L}}(w+\epsilon)$, with $\beta_k \le e^{-k/6}$ replacing the Gaussian tail; *(ii)* the Bernoulli KL replaces the Gaussian KL; and *(iii)* both bounds control the *expected perturbed loss* $\mathbb{E}_{w'\sim Q}[\mathcal{L}(w')]$ rather than the deterministic loss at the center—minimizing $\mathcal{L}_{\mathrm{adv}}(w)$ is motivated by its dominance on the right-hand side in both frameworks. The crucial difference is that our bound operates natively on $\{-1,+1\}^n$.

D.2. Derivation II: Compression-Based Generalization

A second, independent justification comes from a counting argument on the Boolean hypercube. The key insight is that if w has low adversarial loss, then *every* configuration in $\mathcal{B}_k(w)$ also has low loss, and this entire set can be "described" by specifying only w itself.

Theorem D.4 (Hamming-Ball Compression Bound). *Let $\mathcal{H} = \{-1,+1\}^n$ be the hypothesis class, and suppose the BOLD-SAM solution $\hat{w}$ satisfies $\mathcal{L}_{\mathrm{adv}}(\hat{w}) \le \epsilon$ on a training set $\mathcal{D}$ of size m. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the draw of $\mathcal{D}$, $\hat{w}$ satisfies:*

$$\mathcal{L}(\hat{w}) \;\le\; \epsilon + \sqrt{\frac{n\log 2 - \log\sum_{j=0}^{k}\binom{n}{j} + \log\frac{1}{\delta}}{2m}}. \tag{19}$$

Proof. The proof uses a uniform convergence argument over a reduced hypothesis class. By a greedy covering construction, the hypercube $\{-1,+1\}^n$ can be covered by at most $\lceil 2^n/V_k\rceil$ Hamming balls of radius k: repeatedly select an uncovered point as a new center until all points are covered; each ball covers exactly V_k points, so at most $\lceil 2^n/V_k\rceil$ centers suffice.

The BOLD-SAM objective selects the center $\hat{w}$ of a ball, and $\mathcal{L}_{\mathrm{adv}}(\hat{w}) \le \epsilon$ guarantees that every $w' \in \mathcal{B}_k(\hat{w})$ has empirical loss $\le \epsilon$. The effective number of distinguishable hypotheses is therefore at most $\lceil 2^n/V_k\rceil$. By Hoeffding's inequality and a union bound over these centers: for any fixed center w, $\Pr\big[|\mathcal{L}(w) - \hat{\mathcal{L}}(w)| > t\big] \le 2\exp(-2mt^2)$. Taking a union bound and solving for the confidence level δ:

$$\mathcal{L}(\hat{w}) \;\le\; \epsilon \;+\; \sqrt{\frac{\log\lceil 2^n/V_k\rceil + \log(2/\delta)}{2m}} \;\le\; \epsilon \;+\; \sqrt{\frac{n\log 2 - \log V_k + \log(2/\delta)}{2m}}, \tag{20}$$

where we used $\hat{\mathcal{L}}(\hat{w}) \le \mathcal{L}_{\mathrm{adv}}(\hat{w}) \le \epsilon$ and $\log\lceil 2^n/V_k\rceil \le n\log 2 - \log V_k + 1 \le n\log 2 - \log V_k + \log 2$, absorbing the constant into the $\log(2/\delta)$ term. □

The term $\log V_k = \log\sum_{j=0}^{k}\binom{n}{j}$ acts as a "complexity reduction" that quantifies the benefit of Hamming-ball flatness. For $k = \Theta(\sqrt{n})$, this scales as $\Theta(\sqrt{n}\log n)$, providing substantial reduction over the raw $n\log 2$ complexity.

Corollary D.5 (BOLD-SAM Reduces Effective Hypothesis Complexity)**.** *Minimizing $\mathcal{L}_{\mathrm{adv}}(w)$ with Hamming radius k reduces the effective log-cardinality of the hypothesis class from $n\log 2$ to $n\log 2 - \log\sum_{j=0}^{k}\binom{n}{j}$. For $k \ll n$, the reduction is approximately $k\log(n/k) - (k/2)\log(2\pi k)$ bits (by Stirling's approximation of $\binom{n}{k}$).*

D.3. Derivation III: Distributionally Robust Optimization

The min-max objective (1) admits a natural interpretation as a Distributionally Robust Optimization (DRO) problem.

Proposition D.6 (BOLD-SAM as DRO)**.** *The BOLD-SAM objective is equivalent to a DRO problem over discrete ambiguity sets:*

$$\mathcal{L}_{\mathrm{adv}}(w) = \max_{w'\in\mathcal{B}_k(w)} \hat{\mathcal{L}}(w') = \max_{\mu\in\Delta(\mathcal{B}_k(w))} \mathbb{E}_{w'\sim\mu}\big[\hat{\mathcal{L}}(w')\big], \tag{21}$$

where $\Delta(\mathcal{B}_k(w))$ denotes the set of all probability distributions supported on $\mathcal{B}_k(w)$. Moreover, the maximizing distribution μ^* is the Dirac measure $\mu^* = \delta_{w^*}$ where $w^* = \arg\max_{w'\in\mathcal{B}_k(w)} \hat{\mathcal{L}}(w')$.

Proof. For any $\mu \in \Delta(\mathcal{B}_k(w))$, $\mathbb{E}_{w'\sim\mu}[\hat{\mathcal{L}}(w')] \le \max_{w'\in\mathcal{B}_k(w)} \hat{\mathcal{L}}(w') = \mathcal{L}_{\mathrm{adv}}(w)$. Equality is achieved by $\mu^* = \delta_{w^*}$, which is a valid element of $\Delta(\mathcal{B}_k(w))$ since $w^* \in \mathcal{B}_k(w)$. □

This perspective reveals a direct connection to the robustness literature: BOLD-SAM guards against the *worst-case realization* within the ambiguity set $\mathcal{B}_k(w)$, which in the continuous DRO framework corresponds to an ambiguity set defined by a Wasserstein ball, here replaced by its natural discrete counterpart—the Hamming ball. Standard DRO theory [31] guarantees that the solution to (21) provides an upper bound on the population risk under distributional shift, formalized as follows:

Theorem D.7 (DRO Generalization via Hamming Ambiguity Sets)**.** *Let $\hat{w}$ minimize $\mathcal{L}_{\mathrm{adv}}(w)$ over $\{-1,+1\}^n$, and let $\mathcal{L}(w) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[\ell(f_w(x),y)]$ denote the population risk. Suppose the sample loss satisfies $|\hat{\mathcal{L}}(w) - \mathcal{L}(w)| \le \epsilon_m$ uniformly over $\mathcal{B}_k(\hat{w})$ with probability at least $1-\delta$. Then:*

$$\mathcal{L}(\hat{w}) \;\le\; \mathcal{L}_{\mathrm{adv}}(\hat{w}) + \epsilon_m \;\le\; \min_{w\in\{-1,+1\}^n} \mathcal{L}_{\mathrm{adv}}(w) + \epsilon_m. \tag{22}$$

In particular, if there exists a "true" parameter $w^\star$ with $\mathcal{L}(w') \le \epsilon^\star$ for all $w' \in \mathcal{B}_k(w^\star)$, then $\mathcal{L}(\hat{w}) \le \epsilon^\star + 2\epsilon_m$.

Proof. The first inequality is immediate: $\mathcal{L}(\hat{w}) = \hat{\mathcal{L}}(\hat{w}) + [\mathcal{L}(\hat{w}) - \hat{\mathcal{L}}(\hat{w})] \le \hat{\mathcal{L}}(\hat{w}) + \epsilon_m \le \mathcal{L}_{\mathrm{adv}}(\hat{w}) + \epsilon_m$, since $\hat{w} \in \mathcal{B}_k(\hat{w})$. The second follows because $\hat{w}$ minimizes $\mathcal{L}_{\mathrm{adv}}$.

For the "true parameter" claim, note $\mathcal{L}_{\mathrm{adv}}(w^\star) = \max_{w'\in\mathcal{B}_k(w^\star)} \hat{\mathcal{L}}(w') \le \max_{w'\in\mathcal{B}_k(w^\star)} [\mathcal{L}(w') + \epsilon_m] \le \epsilon^\star + \epsilon_m$. Since $\hat{w}$ minimizes $\mathcal{L}_{\mathrm{adv}}$, $\mathcal{L}_{\mathrm{adv}}(\hat{w}) \le \mathcal{L}_{\mathrm{adv}}(w^\star) \le \epsilon^\star + \epsilon_m$. Combining: $\mathcal{L}(\hat{w}) \le \epsilon^\star + 2\epsilon_m$. □

E. Choosing the Hamming Radius k: Full Analysis

The Hamming radius k is the central hyperparameter of BOLD-SAM, controlling both the strength of the adversarial probe and the tightness of the generalization guarantee. We now provide a formal analysis of its effect on the generalization bound, derive the optimal k in closed form for the compression bound, and give practical guidelines.

E.1. The Bias–Sharpness Decomposition

The generalization bounds of § 4 all share a common structure: the population loss is bounded by the sum of a *sharpness term* (which increases with k) and a *complexity term* (which decreases with k). To make this tradeoff explicit, we decompose the compression bound (19) as:

$$\mathcal{L}(\hat{w}) \;\le\; \underbrace{\mathcal{L}_{\mathrm{adv}}(\hat{w};k)}_{\text{sharpness: }\uparrow\text{ in }k} \;+\; \underbrace{\sqrt{\frac{n\log 2 - \log V_k + \log(1/\delta)}{2m}}}_{\text{complexity: }\downarrow\text{ in }k}, \tag{23}$$

where $V_k = \sum_{j=0}^{k} \binom{n}{j}$ is the Hamming ball volume and we write $\mathcal{L}_{\mathrm{adv}}(\hat{w}; k)$ to emphasize the dependence on k.

Small k (under-regularization). When $k = 0$, the Hamming ball is a singleton and $\mathcal{L}_{\mathrm{adv}}(w; 0) = \hat{\mathcal{L}}(w)$—BOLD-SAM reduces to standard empirical risk minimization. The complexity term is $\sqrt{n \log 2/(2m)}$, which is vacuous for $n \gg m$. The optimizer can find sharp minima that happen to have low training loss but generalize poorly.

Large k (over-regularization). As k grows, $\mathcal{L}_{\mathrm{adv}}(w; k)$ increases because the adversary can search over more configurations. In the extreme $k = n$, the Hamming ball is the entire hypercube and $\mathcal{L}_{\mathrm{adv}}(w; n) = \max_{w'} \hat{\mathcal{L}}(w')$ for all w—the objective becomes constant and uninformative. The complexity term vanishes ($\log V_n = n \log 2$), but the sharpness term is maximally pessimistic.

E.2. Optimal k from the Compression Bound

To find the value of k that minimizes the right-hand side of (23), we must model how $\mathcal{L}_{\mathrm{adv}}(w; k)$ grows with k. We consider a natural "smoothness" assumption on the discrete loss landscape.

***Definition* E.1** ((L, σ)-Smooth Boolean Landscape)**.** *We say the empirical risk $\hat{\mathcal{L}}$ is (L, σ)-smooth on the Boolean hypercube if, for all $w \in \{-1, +1\}^n$ and all $k \geq 1$:*

$$\mathcal{L}_{\mathrm{adv}}(w; k) - \hat{\mathcal{L}}(w) \;\leq\; L \cdot k^{\sigma}, \tag{24}$$

where $L > 0$ is a scale parameter and $\sigma \in (0, 1]$ controls the rate at which the worst-case loss grows with the Hamming radius.

The exponent σ captures the local geometry: $\sigma = 1$ corresponds to a "rough" landscape where the worst-case loss grows linearly with the number of flipped bits (every additional flip can cause a proportional loss increase). $\sigma \ll 1$ corresponds to a "smooth" landscape where the worst-case loss grows sublinearly (most additional flips hit redundant directions). Empirically, trained neural networks typically exhibit $\sigma \in (0.3, 0.7)$—the loss landscape has diminishing-returns structure due to weight redundancy.

Proposition E.2 (Approximate Optimal k for the Compression Bound)**.** *Under the (L, σ)-smoothness assumption (24) with $k \ll n$, the right-hand side of (23) is approximately minimized at:*

$$k^* \;\approx\; \left(\frac{\log n}{2L\sigma\sqrt{2mn \log 2}} \right)^{1/(\sigma - 1)} \qquad \text{for } \sigma \neq 1,$$

and $k^* \approx n \exp\!\big(-2L\sqrt{2mn \log 2}\big)$ for $\sigma = 1$.

Proof. For $k \ll n$, Stirling's approximation (see, e.g., [9]) gives $\log V_k = \log \sum_{j=0}^{k} \binom{n}{j} \approx k \log(n/k)$ (dominated by the largest term $\binom{n}{k}$). Substituting the smoothness assumption into (23):

$$\mathrm{RHS}(k) \approx \hat{\mathcal{L}}(w) + Lk^{\sigma} + \sqrt{\frac{n \log 2 - k \log(n/k)}{2m}}. \tag{25}$$

Differentiating with respect to k and setting to zero:

$$\frac{d}{dk}\mathrm{RHS}(k) = L\sigma k^{\sigma - 1} - \frac{\log(n/k) - 1}{2\sqrt{2m(n \log 2 - k \log(n/k))}} = 0.$$

For the leading-order behavior (neglecting the -1 and the $k \log(n/k)$ in the denominator relative to $n \log 2$):

$$L\sigma k^{\sigma - 1} \approx \frac{\log(n/k)}{2\sqrt{2mn \log 2}}.$$

Solving for k with $\log(n/k) \approx \log n$ for $k \ll n$:

$$k^{\sigma - 1} \;=\; \frac{\log n}{2L\sigma\sqrt{2mn \log 2}}. \tag{26}$$

For $\sigma \neq 1$, raising both sides to the power $1/(\sigma - 1)$:

$$k^* \approx \left(\frac{\log n}{2L\sigma\sqrt{2mn \log 2}} \right)^{1/(\sigma-1)}. \tag{27}$$

□

For $\sigma > 1$ (superlinear growth), the exponent is positive and k^* is small. For $\sigma \in (0, 1)$ (sublinear growth, the typical empirical regime), the exponent is negative and k^* is larger—reflecting that diminishing-returns landscapes can tolerate broader Hamming-ball probes. For $\sigma = 1$ (linear growth), the equation $k^0 = \text{const}$ is degenerate; direct solution gives $k^* \approx n \exp\big(-2L\sqrt{2mn \log 2}\big)$.

Since L and σ are landscape-dependent quantities that are not known a priori, this formula serves as a qualitative scaling law rather than a numerical prescription. Notably, k^* scales with the network size n (via $\sqrt{n}$ in the denominator), which provides theoretical support for the practical heuristic of setting $k = \lfloor \alpha \cdot n \rfloor$ as a fixed fraction of the total parameters: larger networks benefit from proportionally larger Hamming radii.

E.3. Volume of the Hamming Ball

The combinatorial structure of the Hamming ball plays a central role in the analysis. The following bounds are useful for understanding the behavior of k.

Lemma E.3 (Hamming Ball Volume Bounds)**.** *For $1 \leq k \leq n/2$:*

$$\left(\frac{n}{k}\right)^k \leq V_k = \sum_{j=0}^{k} \binom{n}{j} \leq \left(\frac{en}{k}\right)^k. \tag{28}$$

In particular, $\log V_k = k \log(n/k) + \Theta(k)$.

Proof. The upper bound follows from $\sum_{j=0}^{k} \binom{n}{j} \leq (k+1)\binom{n}{k} \leq (en/k)^k$, using $\binom{n}{k} \leq (en/k)^k$. The lower bound follows from $V_k \geq \binom{n}{k} \geq (n/k)^k$. □

This lemma reveals the *phase transition* in the complexity reduction. For $k = 1$ (single-bit perturbation), the complexity reduction is merely $\log n$ bits—negligible for large models. For $k = \sqrt{n}$, the reduction is $\Theta(\sqrt{n} \log \sqrt{n})$— substantial. For $k = n/2$ (half the bits), $\log V_k \approx n \log 2 - O(\log n)$, nearly eliminating the complexity term. The "sweet spot" lies in the range $k = \Theta(\sqrt{n})$ to $k = \Theta(n^{2/3})$, where the complexity reduction is large enough to matter but the Hamming ball is still small enough that $\mathcal{L}_{\text{adv}}$ remains informative.

E.4. Relationship to Continuous SAM's ρ

The Hamming radius k is the Boolean analogue of the perturbation radius ρ in continuous SAM [10]. The following proposition makes this correspondence quantitative.

Proposition E.4 (Correspondence Between k and ρ)**.** *Consider a continuous weight vector $\tilde{w} \in \mathbb{R}^n$ and its binarization $w = \operatorname{sign}(\tilde{w})$. A perturbation ϵ with $\|\epsilon\|_2 = \rho$ flips, on average,*

$$\mathbb{E}\big[d_H(\operatorname{sign}(\tilde{w}), \operatorname{sign}(\tilde{w} + \epsilon))\big] \;=\; \sum_{i=1}^{n} \Phi\left(\frac{-|\tilde{w}_i|}{\rho/\sqrt{n}}\right), \tag{29}$$

where Φ is the standard normal CDF and ϵ is drawn uniformly on the ℓ_2-sphere of radius ρ.
For the special case of "marginally confident" weights $|\tilde{w}_i| = c$ for all i (constant magnitude), this simplifies to:

$$k_{\text{equiv}} \;=\; n \cdot \Phi\left(\frac{-c\sqrt{n}}{\rho}\right) \;\approx\; n \cdot \frac{\rho}{c\sqrt{2\pi n}} \exp\left(-\frac{c^2 n}{2\rho^2}\right) \tag{30}$$

for $c\sqrt{n} \gg \rho$ (the "confident weight" regime where continuous SAM suffers from the vanishing perturbation effect).

Proof. Each coordinate ϵ_i of a uniform point on the ℓ_2-sphere has marginal distribution approximately $\mathcal{N}(0, \rho^2/n)$ for large n. Bit i flips when $\operatorname{sign}(\tilde{w}_i + \epsilon_i) \neq \operatorname{sign}(\tilde{w}_i)$, which requires $|\epsilon_i| > |\tilde{w}_i|$ with opposite sign. This has probability $\Phi(-|\tilde{w}_i|/(\rho/\sqrt{n}))$. Summing over i and using linearity of expectation gives (29). The Gaussian tail approximation $\Phi(-x) \approx \phi(x)/x$ for $x \gg 1$ yields (30). □

Remark E.5 (The vanishing perturbation effect, quantified). Equation (30) reveals the exponential decay of k_{equiv} in the ratio $c\sqrt{n}/\rho$. For a typical trained network with $c = 2$ (moderate confidence), $n = 10^6$, and the standard SAM choice $\rho = 0.05$: $c\sqrt{n}/\rho = 2000/0.05 = 40{,}000$, giving $k_{\text{equiv}} \approx 0$—continuous SAM flips essentially zero bits. In contrast, BOLD-SAM with $k = 100$ always probes a 100-dimensional subspace. This makes precise the vanishing perturbation argument of § 3.

E.5. Practical Guidelines

Based on the theoretical analysis above and preliminary experiments, we recommend the following approach for setting k:

1. **Scale-invariant default:** Set k as a fraction of the total parameter count: $k = \lfloor \alpha \cdot n \rfloor$ with $\alpha \in [10^{-5}, 10^{-3}]$. The fraction $\alpha = 10^{-4}$ (0.01% of parameters) is a robust choice. For example, a model with $n = 10^6$ parameters, this gives $k = 100$.

2. **Validation-based tuning:** Perform a grid search, e.g., $k \in \{n/100000,\ n/10000,\ n/1000\}$ (i.e., $\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}\}$), monitoring the generalization gap on a held-out set. The optimal k typically lies in the range where the sharpness $B_k = \mathcal{L}_{\text{adv}}(w; k) - \hat{\mathcal{L}}(w)$ is non-trivial (say, $B_k \in [0.01, 0.5]$ for cross-entropy loss) but not so large that the adversarial loss dominates. This is the approach we take in our experiments.

3. **Sharpness-calibrated k:** Monitor B_k during training. If $B_k < 0.01$ persistently, k is too small (the optimizer is not probing enough of the landscape). If $B_k > 1.0$ persistently, k is too large (the adversarial probe is finding configurations far outside the relevant basin). Adjust k to maintain B_k in a moderate range, or use the adaptive schedule of $k(t) = \max(1, \lfloor k_0 e^{-\lambda t} \rfloor)$.