A MIRROR DESCENT FRAMEWORK FOR ADVERSARIAL ATTACKS OVER THE CLIPPED $l_{\infty,1}$ BALL AND GROUP SPARSITY CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a Mirror Descent framework to solve optimization problems over a clipped $\ell_{\infty,1}$ ball, generalizing the ℓ_1-APGD algorithm to group-norm constraints. We resolve the expensive high-dimensional Bregman projection by reducing it to a continuous resource-allocation problem that is computed exactly via a efficient 1D root search. Furthermore, we propose a structurally relaxed projection that decouples within-group coordinate dependencies, reducing computational overhead while systematically enforcing group sparsity. Instantiating this framework as Group-Structured Mirror Descent (GSMD) for adversarial attack, our extensive evaluations on ImageNet reveal a clear trade-off: the exact projection accelerates objective convergence, whereas the relaxation yields strictly sparser modifications. GSMD generates localized, coherent perturbations and achieves state-of-the-art attack success rates on both standard and robust classifiers, requiring significantly fewer backward passes than existing structured attacks.

1 INTRODUCTION

In the evaluation of adversarial robustness, ℓ_1-bounded attacks are crucial because the ℓ_1 distortion accounts for the total variation of the perturbation (Chen et al., 2018). By acting as a convex surrogate for the ℓ_0 metric, this penalty encourages sparsity, forcing the attack to isolate a minimal subset of vulnerable input features (Chen et al., 2018). For practical applications such as image processing, these sparse perturbations must be jointly optimized with strict coordinate-wise box constraints to ensure valid data representations, resulting in a feasible region known as a clipped cross-polytope. However, this standard clipped cross-polytope fails to address the structural reality of visual data, where groups of related features, such as spatial pixels or superpixels, must often be modified jointly rather than independently. Furthermore, state-of-the-art methods for solving these constrained problems generally rely on Projected Gradient Descent (PGD) (Chen et al., 2018; Croce & Hein, 2021), which is suboptimal for the non-Euclidean geometry (Lan, 2020).

In this work, we introduce a Mirror Descent (MD) framework to solve optimization problems over an $\ell_{\infty,1}$ ball-like constraint clipped onto a box. Given a partition of the coordinates into groups $\{I_j\}_{j=1}^k$, we consider the optimization problems over the feasible region

$$\mathcal{C} = \left\{ x \in \mathbb{R}^d : \sum_{j=1}^{k} \max_{i \in I_j} |x_i| \leq D,\ -l_i \leq x_i \leq u_i\ \ \forall i \right\}. \tag{1}$$

Intuitively, this constraint bounds the total variation of the maximal changes within grouped features. Operating over this geometry with Mirror Descent allows us to generalize the ℓ_1-APGD algorithm (Croce & Hein, 2021), which is exactly recovered in the Euclidean, singleton-group limit. While this framework establishes the structural budget, we further introduce a specific projection method based on this geometry to explicitly ensure group sparsity. By decoupling group dependencies, our method applies a strict cutoff to the group-wise maximums, driving the optimization to select a minimal subset of active groups and yielding spatially coherent perturbations rather than scattered noise.

MD generalizes projected gradient methods by replacing the Euclidean proximity measure with a Bregman divergence induced by a problem-dependent mirror map, thereby allowing the update

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

geometry to be adapted to non-Euclidean constraint sets (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003; Lan, 2020). This principle has proved particularly useful for sparse and ℓ_1-type domains, where non-Euclidean updates can better reflect the geometry of the constraint set (Shao et al., 2022; Shao, 2024). However, applying MD directly to this structured set introduces a significant computational challenge: the update requires a Bregman projection onto the complex intersection of the mixed-norm ball and the coordinate-wise box constraints. While the exact projection for the Euclidean, singleton-group case was successfully resolved by the ℓ_1-APGD algorithm (Croce & Hein, 2021), the exact Bregman projection for MD over these grouped constraints has not yet been established.

To the best of our knowledge, we are the first to introduce an exact Bregman projection for the clipped $\ell_{\infty,1}$ geometry. We achieve this by reducing the computationally expensive, high-dimensional projection step into a continuous resource-allocation problem, which is then computed exactly through a efficient, one-dimensional root search. Continuous resource allocation has a well-established optimization literature (Patriksson, 2008; Patriksson & Strömberg, 2015); however, the reduced objective arising here is strictly convex but contains finitely many points of non-differentiability induced by the group-wise clipping operations. By explicitly characterizing these breakpoints and sorting the corresponding dual values, we locate the interval containing the optimum and compute the exact projection through a one-dimensional root search.

We further introduce a relaxed projection that retains one dominant coordinate per group when solving the resource-allocation problem and subsequently reconstructs the remaining group coordinates from the corresponding group magnitude. This reduces the main sorting operation from coordinate level breakpoints to the k group representatives and induces a stronger sparsity cutoff. For the mirror update, we use a smoothed entropic potential, connecting the resulting method to exponentiated-gradient updates (Kivinen & Warmuth, 1997; Ghai et al., 2020; Shao et al., 2022).

As a primary application, we instantiate our optimization framework to generate structured sparse adversarial attacks. Unlike prior methods, our approach imposes spatial structure directly through the feasible set: the clipped $\ell_{\infty,1}$ constraint establishes a strict budget on the sum of group-wise perturbation magnitudes. These groups can flexibly correspond to fixed spatial regions or semantic image superpixels (Achanta et al., 2012). We term this resulting algorithm *Group-Structured Mirror Descent* (GSMD). Extensive evaluations on ImageNet, using both standard and adversarially robust classifiers, reveal a trade-off between our two proposed projection mechanisms: while the exact Bregman projection yields faster objective convergence, the structurally relaxed projection enforces significantly stronger group sparsity.

In summary, our contributions are:

- We formulate Mirror Descent over a clipped $\ell_{\infty,1}$ ball. In the Euclidean, singleton-group limit, this formulation exactly recovers the ℓ_1-APGD algorithm of Croce & Hein (2021).
- We derive an exact Bregman projection for separable, symmetric, and strictly convex mirror potentials by reducing the complex high-dimensional projection into a continuous resource-allocation problem, solved exactly via sorting and a one-dimensional root search.
- We introduce an efficient structurally relaxed projection whose primary computational bottleneck scales with the number of groups rather than coordinate-level breakpoints
- We instantiate this framework as Group-Structured Mirror Descent (GSMD). Through comprehensive empirical evaluations on standard and robust ImageNet classifiers, we benchmark GSMD against established sparse and structured attacks, and analyze the trade-off between our exact and relaxed projections.

The rest of this paper is organized as follows. Section 2 reviews related work on sparse and structured adversarial attacks. Section 3 details our proposed Mirror Descent framework, including the derivation of the exact and relaxed Bregman projections, and formally instantiates the GSMD attack. Section 4 presents our comprehensive empirical evaluation across standard and robust ImageNet classifiers. Section 5 concludes the paper with future research directions. Mathematical proofs, algorithm implementations, and further experimental details are deferred to the Appendix.

2 Related Work

Adversarial attacks are widely used to evaluate the robustness of machine-learning models (Madry et al., 2018; Carlini et al., 2019; Croce & Hein, 2020). Over the last few years, sparse adversarial attacks have been extensively studied and usually rely on ℓ_1-based perturbations (Croce & Hein, 2019; 2021; Dong et al., 2020; Modas et al., 2019; Chen et al., 2018; Pintor et al., 2021; Lin et al., 2025). While standard sparse attacks satisfy imperceptibility, they have a fundamental limitation in their structural awareness. Consequently, structured sparse adversarial attacks have emerged, progressing from enforcing basic group sparsity to explicitly optimizing for interpretability, exact sparsity control, and computational efficiency.

Early approaches like StrAttack (Xu et al., 2019) utilized an ADMM-based framework with a sliding mask and an $\ell_{2,1}$ norm penalty to impose group sparsity. SAPF (Fan et al., 2020) advanced this by formulating the attack as a Mixed Integer Programming (MIP) problem that factorizes perturbations into magnitude and binary selection variables for precise control over the number of perturbed pixels. Homotopy-Attack (Zhu et al., 2021) enforces group sparsity via an $\ell_{2,0}$ norm penalty, resulting in low-rank, blur-like distortions that pose a different structural challenge to classifiers. Most recently, GSE (Sadiku et al., 2023) was proposed as a two-phase algorithm combining non-convex regularization with projected Nesterov's accelerated gradient, explicitly aiming to surpass prior techniques in terms of computation speed and visual explainability while minimizing perturbation magnitude.

Our work generalizes the ℓ_1-APGD algorithm (Croce & Hein, 2021). ℓ_1-APGD first showed that projecting onto the intersection of ℓ_1 and ℓ_∞ constraints can be computed exactly rather than approximated (Tramer & Boneh, 2019; Maini et al., 2020). We extend this exact geometric efficiency to group-structured sparsity through the Bregman projection. We achieve this by reformulating the projection as a resource allocation problem. While existing literature primarily addresses projections with differentiable costs (Patriksson, 2008; Patriksson & Strömberg, 2015), our reduced cost function is strictly convex with finitely many points of non-differentiability, a more complex landscape that our algorithm systematically resolves to find the optimal solution.

3 Algorithmic Framework

It is straightforward to verify that the feasible set

$$\mathcal{C} = \left\{ x \in \mathbb{R}^d : \sum_{j=1}^{k} \max_{i \in I_j} |x_i| \leq D,\ -l_i \leq x_i \leq u_i\ \ \forall i \right\} \tag{2}$$

is convex and compact. Given a strictly convex potential function $\Phi : \mathbb{R}^d \to \mathbb{R}$, its gradient $\nabla\Phi$ defines the associated mirror map. For $\delta_t \in \mathcal{C}$, a decent direction g_t and a stepsize $\eta_t > 0$ at iteration t, the MD step is given by

$$\delta_{t+1} = \arg\min_{\delta \in \mathcal{C}} \langle g_t, \delta \rangle + \mathcal{B}_\Phi(\delta, \delta_t), \tag{3}$$

where $\mathcal{B}_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y \rangle$ denotes the Bregman divergence associated with Φ. Computing the Mirror Descent update directly for the structured set $\mathcal{C}$ is non-trivial due to the complex intersection of the group-norm and local box constraints. To handle this efficiently, we consider the equivalent two-step formulation of Mirror Descent

$$\delta_{t+\frac{1}{2}} = \nabla\Phi^*(\nabla\Phi(\delta_t) - \eta_t g_t), \tag{4}$$

$$\delta_{t+1} = \arg\min_{\delta \in \mathcal{C}} \mathcal{B}_\Phi(\delta, \delta_{t+\frac{1}{2}}), \tag{5}$$

where Φ^* denotes the convex conjugate of Φ. The first step in equation 4 is typically straightforward to compute. Therefore, the primary technical challenge, and the focus of the this section, is deriving an efficient algorithm to solve the Bregman projection onto our structured feasible set $\mathcal{C}$, outlined in the second step in equation 5.

3.1 Exact Bregman Projection

To efficiently compute the projection step in our setting, we require the update generating function to have a specific separable structure, described in Assumption 1. Unlike general-purpose convex

Algorithm 1 Bregman Projection on $C\ \Pi_\phi(z, D, l, u, \{I_j\})$

Require: $z \in \mathbb{R}^d, l, u \in \mathbb{R}^d_{\geq 0}$. $D > 0, I_1, \ldots, I_k$

$c_i \leftarrow u_i \mathbf{1}_{\{z_i \geq 0\}} + l_i \mathbf{1}_{\{z_i < 0\}}$ for $i = 1, \ldots, d$

$i_j \in \arg\max\{\min\{c_i, |z_i|\}\}$

$C_j = \{0, \min\{c_i, |z_i|\} | i \in I_j, i \neq i_j\}$

$\mathcal{G} = \{0\} \cup (\bigcup_{j=1}^k \bigcup_{c \in C_j} \{-\Phi'_{j,+}(c), -\Phi'_{j,-}(c)\})$

Sort $\mathcal{G}$ in ascending order

for $g \in \mathcal{G}$ **do**

  if $\sum_{j=1}^k \max\left\{\min\{\nabla\Phi_j^*(-g), c_j\}, 0\right\} \leq D$ **then**

    $A = \{j \in \{1 \ldots k\} \mid g \notin \partial\Phi_j(\min\{c_i, |z_i|\}) \text{ for all } i \in I_j\}$

    $L = \{j \in \{1 \ldots k\} \mid \nabla\Phi_j^*(-g) \leq 0\}$

    $U = \{j \in \{1 \ldots k\} \mid \nabla\Phi_j^*(-g) \geq c_{i_j}\}$

    Define $\Psi : \mathbb{R} \to \mathbb{R}, \lambda \mapsto \sum_{j \in A \setminus U \setminus L} \nabla\Phi_j^*(-\lambda)$

    $\lambda = \Psi^{-1}(D - \sum_{j \notin A} \nabla\Phi_j^*(-g) - \sum_{j \in U} c_{i_j})$ ▷ e.g. by bisection in general case

    $|x_{i_j}| \leftarrow \max\left\{\min\{\nabla\Phi_j^*(-\lambda), c_j\}, 0\right\}$ for $j = 1, \ldots, k$

    $|x_i| \leftarrow \min\{|x_{i_j}|, |z_i|, c_i\}$ for $i \in I_j, j = 1, \ldots, k$

    return $|x| \operatorname{sgn}(z)$

  end if

end for

solvers that scale poorly with dimension, our separable setup reduces the projection step to a single one-dimensional root-finding problem. By leveraging a sorting-based approach to locate the exact dual variable, Algorithm 1 computes the projection efficiently. Proposition 3.1 formally establishes the correctness of this reduction.

Assumption 1. *Let the update generating function $\Phi : \mathbb{R}^d \to \mathbb{R}$ be separable such that $\Phi(x) = \sum_{i=1}^d \phi(x_i)$. We assume that the per-coordinate function $\phi : \mathbb{R} \to \mathbb{R}$ is symmetric, strictly convex, and satisfies $\phi(0) = 0$.*

Proposition 3.1. *Suppose Assumption 1 holds. Given $l, u \in \mathbb{R}^d_{\geq 0}$, $z \in \mathbb{R}^d$, $D > 0$ and $I_1, \ldots, I_k$ a partition of $\{1, \ldots, d\}$, Algorithm 1 returns the solution to the optimization problem*

$$\min_{x \in \mathcal{C}} \quad \sum_{i=1}^d (\phi(x_i) - \phi'(z_i)x_i) \tag{6}$$

PROOF SKETCH

The proof proceeds by exploiting the separability of the objective and the group structure of the constraints to reduce the d-dimensional projection into a simplified k-dimensional resource allocation problem. This is based on the optimality condition described in the following lemma.

Lemma 3.2. *Let x^* be the optimal solution to equation 6.For each i, let $c_i = u_i$ if $z_i > 0$, and $c_i = l_i$ if $z_i \leq 0$. If $i_j \in \arg\max_{l \in I_j} \min\{|z_l|, c_l\}$, then $|x_i^*| = \min\{|z_i|, c_i, |x_{i_j}^*|\}$ holds for all $i \in I_j$.*

Since ϕ is symmetric, x_i^* should have the same sign as z_i. Therefore, we can assume w.l.o.g. $x_i > 0$ and $z_i > 0$. To formally decouple the constraints, we use the optimality conditions to define a condensed cost function for each block. For any group I_j, let $\Phi_j : \mathbb{R} \to \mathbb{R}$ represent the cumulative block cost, defined as:

$$\Phi_j(x) = \phi(x) - \phi'(z_{i_j})x + \sum_{i \in I_j, i \neq i_j} (\phi(\min\{z_i, c_i, x\}) - \phi'(z_i)\min\{z_i, c_i, x\}). \tag{7}$$

The original optimization problem in equation 6 can be reduced to

$$\min_{x \in \mathbb{R}^k, \|x\|_1 \leq D, 0 \leq x_i \leq c_i} \quad \sum_{j=1}^k \Phi_j(x_j). \tag{8}$$

Algorithm 2 Relaxed Bregman Projections of Grouped Features $\tilde{\Pi}_\phi(z, D, l, u, \{I_j \mid 1 \le j \le k\})$

Require: Infeasible Variable: z, Perturbation Budget: $D \in \mathbb{R}_{>0}$, Box Constraints: $l, u \in \mathbb{R}_{\ge 0}$, Partition $\mathcal{I} = \{I_1, \ldots, I_k\}$
Ensure: Projected point x
1: $c \leftarrow u \odot \mathbf{1}_{\{z \ge 0\}} + l \odot \mathbf{1}_{\{z \le 0\}}$
2: $i_j = \arg\max_{l \in I_j} \{\min\{|z_l|, c_l\}\}$ for $j = 1, \ldots, k$
3: $\tilde{z} \leftarrow (z_{i_1}, \ldots, z_{i_k})^\top$
4: $\tilde{x} \leftarrow \Pi_\phi(\tilde{z}, D, l, u, \{\{i_j\} \mid 1 \le j \le k\})$ ▷ Run Algorithm 1 with one variable per group
5: $x_i \leftarrow \min\{|z_i|, c_i, |\tilde{x}_{i_j}|\} \operatorname{sgn}(z_i)$ for $i \in I_j$ and $j = 1, \ldots, k$

Structurally, equation 8 is a standard continuous resource allocation problem. Because of the $\min$ operators in the definition of Φ_j, the objective is a strictly convex function characterized by finitely many non-differentiable points. By sorting these non-differentiable points, we can systematically isolate the optimal interval and solve the entire resource allocation problem exactly via 1-D root finding. Designing an efficient algorithm to navigate this search and compute the exact root constitutes the main technical challenge of our projection framework. The complete description of our method for the allocation problem, along with its rigorous proof of correctness, are deferred to the Appendix.

3.2 Implementation and Relaxation

The reduced problem operates over a clipped cross polytope. Because standard PGD is suboptimal for this geometry, we apply an exponentiated gradient update using the smoothed entropic generating function $\Phi(x) = \sum_{i=1}^{d} \phi(x_i)$, where:

$$\phi : \mathbb{R} \to \mathbb{R}, \quad x \mapsto (|x| + \beta) \ln\left(\frac{|x|}{\beta} + 1\right) - |x|.$$

for a smoothing parameter $\beta > 0$. The complete algorithmic implementation is detailed in the Appendix.

Our fundamental motivation for enforcing an $\ell_{\infty,1}$-norm boundary is to induce hard sparsity by truncating groups that yield a lower optimization payoff. However, the exact projection in equation 7 diminishes this by aggregating all variables within a group. Consequently, extremely small values inside highly active groups limit their budget consumption, causing excess budget to spill over and keep marginal groups alive.

To enforce a more effective cutting mechanism, we leverage the optimality conditions from Lemma 3.2 to decouple these dependencies. We explicitly solve the optimization problem for the dominant variable i_j in each group j, setting the remaining dependent variables in the group accordingly. This powerful reduction not only amplifies the sparsity-inducing cutoff but also allows us to further relax the problem to achieve a closed-form implementation.

Specifically, as detailed in the Appendix, if the partition consists of singleton groups, the exact root of the resource allocation problem admits a strictly analytic solution. Because PGD is a special case of MD under a Euclidean distance metric, operating our framework under this metric with singleton groups allows us to exactly recover the efficient ℓ_1-APGD algorithm (Croce & Hein, 2021). This relaxation is outlined in Algorithm 2.

Finally, the relaxation improves per-iteration complexity. The exact algorithm must evaluate subgradients at $O(d)$ breakpoints—taking $\mathcal{O}(d \max_j |I_j|)$ time. In contrast, our relaxed formulation bypasses these evaluations entirely, reducing the primary computational bottleneck to simply sorting the k active groups.

3.3 Application to Group Structured Adversarial Attack

Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be an input-label pair, where $\mathcal{X} = [0, 1]^d$ is the box-constrained feature space, and let $f : \mathcal{X} \to \mathcal{Y}$ denote a predictive model. We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ and a perturbation vector δ. For a fixed-budget attack, the goal is to find a perturbation that maximizes the model's loss within a restricted structural region.

Given a partition of the features into k disjoint groups $\{I_j\}_{j=1}^k$, a structured attack bounds the mixed $\ell_{\infty,1}$ norm of δ while ensuring the perturbed input $x+\delta$ remains a valid feature vector. By defining the local box constraints as $l_i = x_i$ and $u_i = 1 - x_i$, we obtain the feasible set:

$$\mathcal{C} := \left\{ \delta \in \mathbb{R}^d \mid \sum_{j=1}^{k} \max_{i \in I_j} |\delta_i| \leq D, \quad -l_i \leq \delta_i \leq u_i \quad \forall i \in \{1, \ldots, d\} \right\}.$$

The adversarial perturbation is then found by solving the following constrained optimization problem:

$$\max_{\delta \in \mathcal{C}} \quad \ell(f(x+\delta), y).$$

In this formulation, the threat budget is consumed by the maximum absolute perturbation within each group. By penalizing the ℓ_1-norm of these group-wise ℓ_∞-norms, the constraint explicitly curtails cross-group variation. Once a group I_j is selected for modification—indicated by a non-zero value for $\max_{i \in I_j} |\delta_i|$—the optimization can freely distribute the perturbation across all features within that group up to that maximum threshold.

The flexibility of the feature groups $\{I_j\}$ allows this framework to recover several common attack paradigms and adapt to different semantic structures:

- *Pixel-wise Sparsity* (ℓ_1 Attack): If each group contains a single feature ($|I_j| = 1$), the constraint reduces to the standard ℓ_1-norm, $\sum_{i=1}^{d} |\delta_i| \leq D$.
- *Color-Channel Grouping*: By grouping the channels of each pixel, the attack enforces sparsity in pixel locations rather than individual color values.
- *Spatial/Semantic Grouping*: Features can be organized into k spatial grids or superpixels. This forces the attack to select a minimal subset of spatial regions while allowing complex modifications within each active segment.

While Mirror Descent (MD) with the exact Bregman projection can solve this optimization problem, we specifically seek to enforce a stricter sparsity cutoff. To prioritize group-level sparsity and isolate a minimal subset of active groups, we instantiate our framework using MD with the structurally relaxed projection. We term this approach *Group-Structured Mirror Descent* (**GSMD**):

$$\nabla\Phi(\delta_{t+\frac{1}{2}}) = \nabla\Phi(\delta_t) + \eta_t \nabla l(f(x+\delta_t), y)$$
$$\delta_{t+1} = \tilde{\Pi}_\phi(\delta_{t+\frac{1}{2}}, D, l, u, \{I_j \mid 1 \leq j \leq k\}).$$

By applying this formulation, GSMD effectively maintains a uniform threat model while driving the attack toward localized, spatially coherent modifications.

4 Empirical Evaluation

In this section, we comprehensively evaluate our proposed framework [1]. We first conduct a comparison to validate the operational advantages of our structural relaxation. Subsequently, we benchmark this relaxed formulation against state-of-the-art structured sparse attacks across both standard and adversarially trained classifiers. We evaluate our approach on 10,000 ImageNet validation images across standard (ResNet50, EfficientNetB3, Vit_B_16) and adversarially robust classifiers. We compare two variants of our method—**GSMD-f** (fixed grid) and **GSMD-s** (SLIC superpixels)—against standard structured sparse attacks, including StrAttack, FWnucl, and GSE. Following the procedure from (Sadiku et al., 2023), we select ten target labels for each image according to $t_j^{(i)} = (y_i + a_j) \bmod 1000$, where y_i is the true label and $a_j \in \{1, \ldots, 1000\}$ is randomly chosen. To accurately reflect structural objectives, we report the best, average, and worst results across spatial coherence (ANC), patch sparsity ($d_{2,0}$), average perturbed pixels (ACP), and our strict sparsity-aware Attack Success Rate (ASR_τ).

[1] Code is made available at `https://anonymous.4open.science/r/GSMD-273A`.

4.1 Comparison: Exact vs. Relaxed Projection

We empirically demonstrate that leveraging the reduction to a partition of singleton groups improves sparsity over the exact projection while maintaining comparable convergence, see Figure 1. We conduct untargeted attacks on ResNet50 for 100 randomly selected images from ImageNet. While

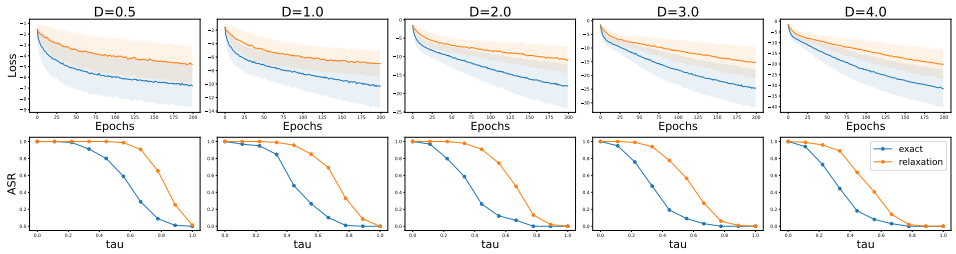


Figure 1: First row shows convergence of GSMD for the exact projection Algorithm 1 and the relaxed version outlined in Algorithm 2. The second row displays the adjusted average success rates (9). We report both metrics for $D \in \{0.5, 1, 2, 3, 4\}$.

the exact projection algorithm demonstrates faster optimization convergence in the loss landscape, the relaxed formulation proves superior at enforcing structured sparsity. By decoupling the group dependencies and applying a stricter cutoff to marginal features, the relaxation maintains a significantly higher adjusted average success rate (ASR_τ) across stricter sparsity thresholds. Given its superiority in inducing hard sparsity and its reduced per-iteration computational cost, we employ the relaxed GSMD formulation (with both fixed and superpixel grids) for all subsequent comparisons against state-of-the-art methods.

4.2 Targeted Attacks on Standard Classifiers

We first evaluate targeted attacks against standard ImageNet classifiers. Table 1 reports results on ResNet50, EfficientNet-B3, and ViT-B-16 across best-, average-, and worst-case settings.

While several methods reach 100% $\text{ASR}_{0.0}$, their performance separates under stricter sparsity constraints. On ResNet50, GSMD achieves 99%, 76%, and 27% $\text{ASR}_{0.8}$ in the best, average, and worst cases, compared with 79%, 33%, and 4% for the strongest baseline. On EfficientNet-B3, GSMD reaches 100%, 96%, and 73%, compared with 98%, 63%, and 26%. Thus, GSMD preserves substantially higher attack success as the sparsity requirement tightens.

The perturbation metrics distinguish coordinate from structural sparsity. While ℓ_1-APGD often achieves lower ACP, GSMD concentrates perturbations into fewer spatial regions, yielding lower ANC and $d_{2,0}$. For average-case ResNet50, GSMD attains ANC 2.70 and $d_{2,0} = 7{,}569$, compared with ANC 7.58 and $d_{2,0} = 26{,}591$ for FWnucl and $d_{2,0} = 10{,}785$ for ℓ_1-APGD. Similar trends hold on EfficientNet-B3. Overall, the mixed $\ell_{\infty,1}$ constraint promotes group-level sparsity rather than minimizing the number of changed coordinates alone. Figure 2 reports sparsity-aware attack success over $\tau \in [0, 1]$.

4.3 Targeted Attacks on Adversarially Trained Classifiers

We next evaluate GSMD against adversarially trained and robust ImageNet classifiers. Table 2 reports results for two robust ResNet50 models and ConvNeXt-S.

GSMD achieves 100% $\text{ASR}_{0.0}$ across all reported settings, with larger differences emerging under structural sparsity constraints. On the two robust ResNet50 models, GSMD reaches up to $100/80/32\%$ and $100/82/33\%$ $\text{ASR}_{0.5}$ in the best/average/worst cases, compared with $83/37/11\%$ and $81/36/10\%$ for the strongest baselines. On ConvNeXt-S, GSMD reaches 100%, 100%, and 92%, compared with 100%, 94%, and 76% for StrAttack. Thus, GSMD retains substantially higher attack success as the structural sparsity constraint tightens.

GSMD also achieves the lowest ANC and $d_{2,0}$ across the reported robust-model settings, indicating more spatially concentrated perturbations. On ConvNeXt-S in the average case, GSMD-s attains

Table 1: Targeted attacks performed against ResNet50, EfficientNet_B3, and ViT-B-16 classifier for ImageNet. Tested on 10k images from each dataset, and 10 target labels for ImageNet. The average improvement is calculated with respect to the best baseline method for each metric and case.

		Best case					Average case					Worst case				
	Attack	**ASR$_{0.0}$**	**ASR$_{0.8}$**	**ACP**	**ANC**	$d_{2,0}$	**ASR$_{0.0}$**	**ASR$_{0.8}$**	**ACP**	**ANC**	$d_{2,0}$	**ASR$_{0.0}$**	**ASR$_{0.8}$**	**ACP**	**ANC**	$d_{2,0}$
ResNet50	FWnucl	31.1%	19%	9,897	3.81	11295	7.34%	3.0%	19,356	**7.58**	26,591	0.0%	0.0%	N/A	N/A	N/A
	StrAttack	**100%**	56%	6,579	7.18	9,620	**100%**	16%	15,071	18.0	20,921	**100%**	1.0%	26,908	32.1	34,768
	ℓ_1-APGD	**100%**	79%	**494**	191	5,214	**100%**	33%	**1,868**	434	10,785	100%	0.3%	**5,224**	788	18,902
	GSE	**100%**	77%	3,516	5.89	5,967	**100%**	31%	12,014	14.6	16,724	**100%**	4.0%	21,675	22.8	29,538
	GSMD-f (Ours)	**100%**	**99%**	**2,973**	1.31	**3,832**	**100%**	70%	**6,260**	3.48	**7,569**	**100%**	21%	**11,836**	6.83	**13,552**
	GSMD-s (Ours)	**100%**	**99%**	3,100	**1.14**	4,157	**100%**	**76%**	6,505	**2.70**	8,190	**100%**	**27%**	11,842	**5.51**	14,235
EfficientNet	FWnucl	44%	10%	4,392	2.12	5,445	10%	9.0%	5,299	3.51	6,719	0.0%	0.0%	N/A	N/A	N/A
	StrAttack	**100%**	79%	5,727	6.44	8,467	**100%**	47%	15,470	23.3	22,987	**100%**	8.0%	30,460	46.7	43,831
	ℓ_1-APGD	**100%**	98%	**476**	190	3,708	**100%**	63%	**1,424**	416	8,767	**100%**	11%	**3,884**	829	18,437
	GSE	**100%**	74%	2,700	7.27	4,988	**100%**	49%	8,881	17.2	13,745	**100%**	26%	19,401	29.4	26,728
	GSMD-f (Ours)	**100%**	**100%**	1,828	**1.06**	2,353	**100%**	**96%**	**4,080**	**2.17**	**5,039**	**100%**	**73%**	**8,330**	**4.79**	**10,027**
	GSMD-s (Ours)	**100%**	**100%**	**1,618**	**1.04**	**2,346**	**100%**	94%	4,910	2.37	6,432	**100%**	64%	9,399	5.09	12,225
ViT_B_16	FWnucl	31.1%	19%	9,897	3.81	11295	7.34%	3.0%	19,356	**7.58**	26,591	0.0%	0.0%	N/A	N/A	N/A
	StrAttack	**100%**	56%	6,579	7.18	9,620	**100%**	16%	15,071	18.0	20,921	**100%**	1.0%	26,908	32.1	34,768
	ℓ_1-APGD	**100%**	87%	**260**	51	**2,270**	**100%**	37%	**1,400**	144	**6,483**	**100%**	2.1%	**5,126**	355	15,989
	GSE	**100%**	77%	3,516	5.89	5,967	**100%**	31%	12,014	14.6	16,724	**100%**	4.0%	21,675	22.8	29,538
	GSMD-f (Ours)	**100%**	**99%**	2,973	1.31	3,832	**100%**	70%	6,260	3.48	7,569	**100%**	21%	11,836	6.83	**13,552**
	GSMD-s (Ours)	**100%**	**99%**	3,100	**1.14**	4,157	**100%**	**76%**	6,505	**2.70**	8,190	**100%**	**27%**	11,842	**5.51**	14,235

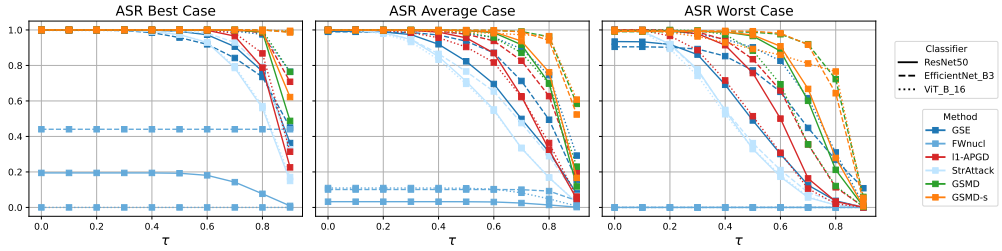


Figure 2: Shows the adjusted average success rate (9) for targeted adversarial attacks on ResNet50, EfficientNet, and ViT_B_16 for ImageNet (color indicates method, and line styles corresponding classifier).

100% ASR$_{0.5}$ with ANC 4.03 and $d_{2,0} = 5{,}671$, while GSMD-f reaches 99% with $d_{2,0} = 3{,}423$. StrAttack reaches 94% with ANC 14.7 and $d_{2,0} = 12{,}196$. While ℓ_1-APGD can achieve lower ACP, its larger ANC and $d_{2,0}$ reflect weaker structural sparsity. Overall, GSMD combines high sparsity-aware attack success with perturbations concentrated in fewer spatial regions. Figure 3 reports results over $\tau \in [0, 1]$.

Additionally, GSMD consistently shows the best ASR for various thresholds $\tau \in [0, 1.0]$ illustrated in Figure 3.

5 Conclusion

In this work, we introduced a Mirror Descent framework to solve optimization problems over a clipped $\ell_{\infty,1}$-ball, providing a generalization of the ℓ_1-APGD algorithm (Croce & Hein, 2021) to group-norm constraint geometries. By reformulating the expensive high-dimensional Bregman projection as a continuous resource-allocation problem, we derived an exact projection operator that can be solved via sorting and a fast 1D root search. Furthermore, we developed a structurally relaxed projection that decouples coordinate dependencies within groups, reducing computational overhead while enforcing a strict cutoff to guarantee group-level sparsity.

Instantiating this framework for adversarial robustness gave rise to Group-Structured Mirror Descent (GSMD). Our empirical evaluation on ImageNet demonstrated a clear trade-off: while the exact Bregman projection achieves faster objective convergence, the structurally relaxed projection yields strictly sparser, spatially coherent modifications. Leveraging this relaxation, GSMD consistently matched or exceeded state-of-the-art attack success rates under strict sparsity thresholds across both

Table 2: Targeted attacks performed against adversarially trained ResNet50 classifiers, and a robust ConvNeXt-S classifier for ImageNet. Tested on 10k images from each dataset, and 10 target labels for ImageNet.

		Best case					Average case					Worst case				
	Attack	**ASR$_{0.0}$**	**ASR$_{0.5}$**	**ACP**	**ANC**	$d_{2,0}$	**ASR$_{0.0}$**	**ASR$_{0.5}$**	**ACP**	**ANC**	$d_{2,0}$	**ASR$_{0.0}$**	**ASR$_{0.5}$**	**ACP**	**ANC**	$d_{2,0}$
ResNet50	FWnucl	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	StrAttack	**100%**	62%	21,072	4.05	27,597	**100%**	30%	33,779	11.4	39,9914	**100%**	11%	47,537	21.7	51,514
	ℓ_1-APGD	**100%**	83%	5,498	206	18,024	**100%**	37%	15,909	438	34,084	96%	9.0%	26,354	739	48,507
	GSE	**100%**	10%	54,058	11.2	52,985	84%	8.0%	54,047	13.4	51,419	34%	3.0%	65,532	14.7	62,001
	GSMD-f (Ours)	**100%**	97%	4,065	2.61	**6,147**	**100%**	**80%**	**13,850**	7.07	**17,509**	**100%**	**32%**	**29,988**	13.6	**34,702**
	GSMD-s (Ours)	**100%**	**100%**	**4,024**	**1.63**	6,933	**100%**	78%	14,690	**4.91**	19,691	**100%**	28%	31,492	**10.1**	37,547
ResNet50_2	FWnucl	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	StrAttack	**100%**	48%	26,1812	3.13	32,684	**100%**	23%	37,990	9.12	43,160	**100%**	10%	49,446	18.5	52,227
	ℓ_1-APGD	**100%**	81%	6,461	218	20,059	**100%**	36%	16,458	450	34,720	97%	10%	26,461	769	47,563
	GSE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	GSMD-f (Ours)	**100%**	**100%**	**4,955**	2.21	**7,300**	**100%**	**82%**	**14,115**	6.58	**17,793**	**100%**	**33%**	**29,610**	13.2	**34,173**
	GSMD-s (Ours)	**100%**	**100%**	**4,955**	**1.43**	8,205	**100%**	81%	14,690	**4.45**	19,573	**100%**	**32%**	30,395	**9.44**	36,369
ConvNeXt-S	FWnucl	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	StrAttack	**100%**	**100%**	3,884	7.6	6,944	**100%**	94%	7,302	14.7	12,196	**100%**	76%	12,690	23.2	19,558
	ℓ_1-APGD	**100%**	98%	**630**	109	4,602	**100%**	92%	**2,117**	220	9,301	98%	66%	5,899	379	17,213
	GSE	36%	0.0%	43,620	2.6	42,333	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	GSMD-f (Ours)	**100%**	**100%**	1,763	2.3	2,817	**100%**	**99%**	3,891	5.78	3,423	**100%**	90%	8,400	10.8	11,324
	GSMD-s (Ours)	**100%**	**100%**	**1,550**	**1.67**	**2,790**	**100%**	**100%**	**3,423**	**4.03**	**5,671**	**100%**	**92%**	**7,289**	**7.56**	**10,816**

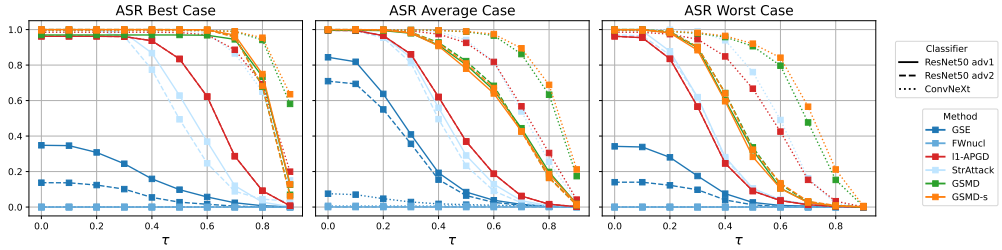


Figure 3: Shows the adjusted average success rate (9) for targeted adversarial attacks on adversarially trained ResNet50, and ConvNeXt for ImageNet (color indicates method, and line styles corresponding classifier).

standard and adversarially trained classifiers, while modifying substantially fewer spatial regions and requiring fewer backward passes than existing baselines.

Despite its strong empirical performance, our framework currently relies on external segmentation methods to define the group partitions. Because the structural boundaries of the attack are dictated entirely by these given tools, the resulting groups may not always align perfectly with the true semantic boundaries of the objects. Consequently, the adaptivity of the attack is constrained, often requiring the manual selection of an appropriate segmenter to achieve optimal application.

Future research will focus on overcoming these structural limitations by integrating learned, dynamic grouping strategies that can explicitly adapt to and align with semantic object structures during the optimization process. We also aim to generalize the GSMD framework to other data modalities, such as Natural Language Processing and audio. Finally, exploring robustness certification methods under structured sparsity constraints and developing specific defenses tailored against coherent, group-structured perturbations remain critical next steps for the community.

AI USE STATEMENT

In this work, we used generative AI tools for text editing, grammatical refinement, and prototyping implementation code. We have not used generative AI tools for conceiving research ideas, formulating hypotheses, or deriving mathematical proofs; all theoretical propositions, lemmas, and theorems were formulated and proven solely by the authors. Other tasks involving automated generation of conceptual claims or literature discovery are not applicable to this work.

Additionally, we used generative AI tools for assisting with drafting and refining experimental visualization figures. All written text was originally drafted by the authors, with generative AI serving

exclusively as an editorial tool to enhance conciseness, formatting, and stylistic clarity. All AI-assisted code was manually inspected, debugged, validated, and executed by the authors to guarantee computational correctness and reproducibility across all benchmark runs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure complete empirical and theoretical reproducibility, all mathematical claims, formal propositions, and proofs are detailed with their foundational assumptions in section D. Comprehensive descriptions of experimental protocols, optimization hyperparameter selections, and baseline implementations are provided in section B.1 All experiments rely on publicly available datasets, models, and baseline attacks. An anonymized code repository containing the complete implementation of the considered experimental setup accompanied by clear specification of usage and all dependencies, including external libraries, is available at `https://anonymous.4open.science/r/GSMD-273A` and will be made publicly available upon publication.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11302. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11302`.

Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4724–4732, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Francesco Croce and Matthias Hein. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2201–2211. PMLR, 18–24 Jul 2021.

Francesco Croce and Matthias Hein. Adversarial robustness against multiple and single l_p-threat models via quick fine-tuning of robust classifiers. In *International Conference on Machine Learning*, pp. 4436–4454. PMLR, 2022.

Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. *Advances in Neural Information Processing Systems*, 33:11226–11236, 2020.

Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *European conference on computer vision*, pp. 35–50. Springer, 2020.

Udaya Ghai, Elad Hazan, and Yoram Singer. Exponentiated gradient meets gradient descent. In Aryeh Kontorovich and Gergely Neu (eds.), *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pp. 386–407. PMLR, 08 Feb–11 Feb 2020. URL https://proceedings.mlr.press/v117/ghai20a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ehsan Kazemi, Thomas Kerdreux, and Liqiang Wang. Minimally distorted structured adversarial attacks. *International Journal of Computer Vision*, 131(1):160–176, 2023.

Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and computation*, 132(1):1–63, 1997.

Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.

Fudong Lin, Jiadong Lou, Hao Wang, Brian Jalaian, and Xu Yuan. Towards interpretable adversarial examples via sparse adversarial attack. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 92–110. Springer, 2025.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pp. 6640–6650. PMLR, 2020.

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9087–9096, 2019.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Michael Patriksson. A survey on the continuous nonlinear resource allocation problem. *European Journal of Operational Research*, 185(1):1–46, 2008.

Michael Patriksson and Christoffer Strömberg. Algorithms for the continuous nonlinear resource allocation problem—new implementations and numerical studies. *European Journal of Operational Research*, 243(3):703–722, 2015.

Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. *Advances in Neural Information Processing Systems*, 34:20052–20062, 2021.

Shpresim Sadiku, Moritz Wagner, and Sebastian Pokutta. Gse: Group-wise sparse and explainable adversarial attacks. *arXiv preprint arXiv:2311.17434*, 2023.

Weijia Shao. Improved dimensionality dependence for zeroth-order optimisation over cross-polytopes. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44413–44428. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/shao24a.html.

Weijia Shao, Fikret Sivrikaya, and Sahin Albayrak. Optimistic optimisation of composite objective with exponentiated update. *Machine Learning*, pp. 1–46, August 2022. ISSN 1573-0565. doi: 10.1007/s10994-022-06229-1. URL https://doi.org/10.1007/s10994-022-06229-1. Special Issue of the ECML PKDD 2022 Journal Track.

Naman Deep Singh, Francesco Croce, and Matthias Hein. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. *Advances in Neural Information Processing Systems*, 36:13931–13955, 2023.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32, 2019.

Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR, 2019.

Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *International Conference on Machine Learning*, pp. 12868–12877. PMLR, 2021.

A Appendix

B Experimental Setup

In this section, we describe the datasets, outline the methods, and the classifier used in our study. Additionally, we cover standard metrics and suggest using a sparsity-aware attack success rate (ASR) to appropriately quantify whether an adversarial perturbation is successful with respect to a group-induced sparsity level. In the second part, we lay out the hyperparameters used in our study. All experiments are run onto a system equipped with a NVIDIA RTX 6000 ADA 48GB.

B.1 Datasets, Methods and Metrics

Attacks: We follow the experimental setup described in Sadiku et al. (2023) and investigate the effectiveness and sparsity of our proposed structured sparse attack against SOTA structured sparse adversarial attacks, including *FWnucl* (Kazemi et al., 2023), *StrAttack* (Xu et al., 2019), and *GSE* (Sadiku et al., 2023).

Dataset and Classifiers: We employ a ResNet50 (He et al., 2016), and EfficientNetB3 (Tan & Le, 2019) models on 10k randomly sampled images from the validation set of the ImageNet dataset. Additionally, we consider two adversarially trained ResNet50 models for ImageNet. The first version, denoted ResNet50_adv1 was trained using the setup introduced in Madry et al. (2018) which is robust against adversarial examples within an ℓ_2-ball with radius $\epsilon = 3.0$. The second robust model is a ℓ_2 robust model which is fine-tuned against ℓ_∞ and ℓ_1 threats as proposed in Croce & Hein (2022), which we denote as ResNet50_adv2. Moreover, we deploy an isotropic ConvNeXt with Convolution Stem (ConvNext-iso-CvSt), which was introduced by Singh et al. (2023), which is a specific instance of a ConvNeXt (Liu et al., 2022) classifier with an isotropic design and a convolutional stem, which was tailored against ℓ_∞ threats.

Metrics: Recent studies on structured sparse attacks focus on the attack success rate (ASR). While this is an appropriate metric for adversarial attacks in general, it does not fully reflect their intended objectives: sparsity and structure. Therefore, we propose computing the attack success rate as a function of sparsity. Let $x + \delta \in \mathcal{X}$ denote the adversarial example, where δ is a carefully crafted perturbation and let $\mathcal{G} = \{G_1, \ldots, G_K\}$ be the partition of the spatial domain of x into k disjoint superpixel regions, where each set G_k is a set of pixel coordinates (i, j). We denote the *Group*

activity with

$$A_k(\delta) = \begin{cases} 1 & \text{if } ||M_k \odot \delta||_2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\odot$ is the element-wise multiplication and M_k is a binary mask of the same spatial dimension with respect to the pixel coordinates of group G_k. We compute the normalized group sparsity as

$$\sigma(\delta) = \frac{1}{K} \sum_{k=1}^{K} (1 - A_k(\delta))$$

which is 1 iff. $\delta \equiv 0$ and has a minimal value of 0 if there is at least one non-zero perturbation within each group. The adjusted average success rate for the given group structure and sparsity threshold $\tau \in [0, 1]$ is

$$\text{ASR}_\tau = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[f(x_i + \delta_i) \neq y_i \wedge \sigma(\delta_i) \geq \tau], \tag{9}$$

where y_i is the target label for instance x_i, clearly for $\tau = 0$ is the standard average success rate, such that for any $\tau > 0$ it holds $\text{ASR}_\tau \leq \text{ASR}_0$. Hence, it is a stricter measure that additionally penalizes dense perturbations, and thus, it is a more suitable metric for structured sparse adversarial attacks.

Additionally, we follow the evaluation framework from Sadiku et al. (2023) and consider

$$d_{2,0}(\delta) := |\{i : A_k(\delta) > 0,\ i = 1, \ldots, K\}|, \tag{10}$$

where $\mathcal{G} = \{G_1, \ldots, G_K\}$ is the index sets of all overlapping 8×8 patches of δ, and M_k is the associated binary mask of G_k with the same spatial dimension as δ.

To measure spatial coherence, we follow the procedure from Sadiku et al. (2023) and compute the average number of clusters (ANC) as follows. For each perturbation, a binary mask is created indicating the perturbed pixels, and we interpret it as a graph in which nodes correspond to perturbed pixels and edges connect 4-neighboring pixels. The number of clusters equals the number of connected components in this graph, computed via depth-first search. ANC is the average number of such components across all successful adversarial examples. In Figure 4, we illustrate grid segmentation and superpixel-based segmentation.



(a) Equidistant groups (b) Superpixel groups

Figure 4: Shows two ante-hoc strategies to group pixels. The approach on the left-hand side (a) uses an equidistant grid and (b) relies on the segmentation based on superpixels derived with an independent segmentation method.

B.2 Parameter setting

We follow the attack configuration outlined in Sadiku et al. (2023). Specifically, we use the same optimized hyperparameters for GSE, namely $q = 0.9, \sigma = 0.05, \mu = 0.1, \hat{k} = 50$, for ImageNet and a non-adversarially trained classifier. For robust classifiers, the optimized hyperparameters are set to $q = 0.9, \sigma = 0.75, \mu = 5 \times 10^{-4}, \hat{k} = 150$ accordingly. For StrAttack Xu et al. (2019), we use the recommended hyperparameters, and for FWnucl, we use $\epsilon = 5.0$ for a non-robust classifier, and use $\epsilon = 30$ for an adversarially trained classifier. We use $D = 2.0$ and $\rho = 0.1$ in all attacks on a non-robust classifier, and set $D = 100$ and $\rho = 0.01$ for attacks on an adversarially trained classifier for ImageNet. We use a fixed grid of 15×15 equidistant groups, which we denote as GSMD-f, and use SLIC superpixel segmentation (Achanta et al., 2012) with a maximum of 225 segments and

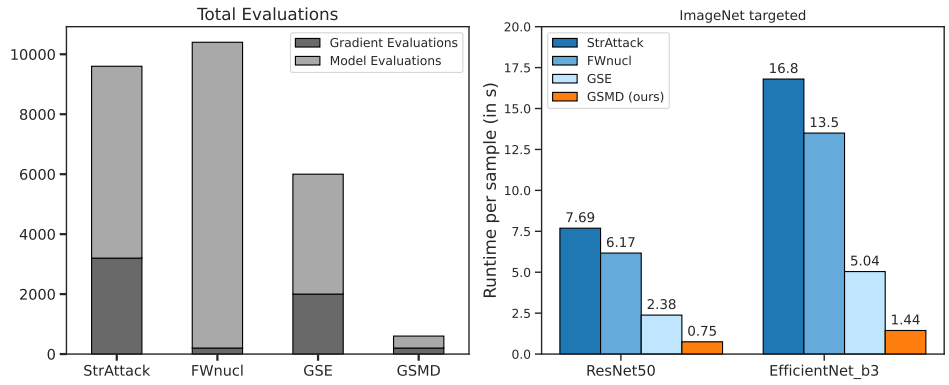


Figure 5: Shows the total number of forward passes and backward passes, and the runtime per adversarial example (in s) for 200 iterations for each method, respectively.

$\sigma = 5.0$. We write GSMD-s for the adversarial attack with groups derived from SLIC segmentation. We run all experiments for a total of 200 iterations.

As outlined, we utilize the smoothed entropic generating function $\Phi(x) = \sum_{i=1}^{d} \phi(x_i)$, where:

$$\phi : \mathbb{R} \to \mathbb{R}, \quad x \mapsto (|x| + \beta) \ln \left(\frac{|x|}{\beta} + 1 \right) - |x|.$$

for a smoothing parameter $\beta > 0$. For all experiments, we fix $\beta = 1.0$. We use an adaptive stepsize which is proportional to inverse of the square root of the accumulated maximum squared gradients up to t, $\eta_t \propto (\sum_{s=1}^{t} \|g_s\|_\infty^2)^{-\frac{1}{2}}$.

C Additional Experiments

C.1 Efficiency and speed comparison

Our GSMD has the fewest backward passes and notably fewer forward passes across all baseline methods. While forward passes may be neglected, even though they contribute to the attack method's total runtime, backward passes carry the most information for updating the perturbation. This means our proposed attack framework is far more efficient in terms of the information required to craft adversarial perturbations. Particularly in our evaluation framework, GSMD needs up to 93% fewer backward passes to achieve superior results compared to StrAttack, as shown in Figure 5.

C.2 Untargeted adversarial attacks

Overall, GSMD consistently matches or exceeds full targeted success while substantially improving sparsity and maintaining the most structured perturbations, establishing it as the strongest method across both architectures, see Table 3.

GSMD achieves 97–98% $\text{ASR}_{0.8}$, substantially outperforming StrAttack (49%) and FWnucl (7.8%), and improving over GSE (85%), producing structured and sparse perturbations. While GSE achieves the lowest ACP (1,624), GSMD achieves competitive ACP (1,930–2,024) but significantly yields the lowest $d_{2,0}$ (2,963) and the lowest ANC (2.80–3.27), indicating more sparse and structured perturbations.

On EfficientNet-B3, GSMD again achieves 100% $\text{ASR}_{0.0}$ and improves $\text{ASR}_{0.8}$ to 98–100%, compared to 83% for GSE and 66% for StrAttack. Simultaneously, GSMD-s attains the lowest ACP (1,640), lowest ANC (1.70), and lowest $d_{2,0}$ (2,556), indicating both higher attack strength and improved sparsity.

Table 3: Untargeted attacks performed against ResNet50, and EfficientNet_B3 calssifier for ImageNet. Tested on 1k images for ImageNet.

	Attack	$ASR_{0.0}$	$ASR_{0.8}$	ACP	ANC	$d_{2,0}$
ResNet50	FWnucl	47%	7.8%	13,807	3.84	16,414
	StrAttack	**100%**	49%	7,692	14.6	12,230
	GSE	97%	85%	**1,624**	8.31	3,430
	GSMD-f (Ours)	**100%**	97%	1,929	3.27	**2.963**
	GSMD-s (Ours)	**100%**	**98%**	2,024	**2,80**	3,301

	Attack	$ASR_{0.0}$	$ASR_{0.8}$	ACP	ANC	$d_{2,0}$
EfficientNet_B3	FWnucl	65%	58%	5,583	2.58	7,193
	StrAttack	**100%**	66%	5,980	14.6	10,219
	GSE	93%	83%	2,013	8.91	4,028
	GSMD-f (Ours)	**100%**	98%	2,007	1.94	2,804
	GSMD-s (Ours)	**100%**	**100%**	**1,640**	**1.70**	**2,556**

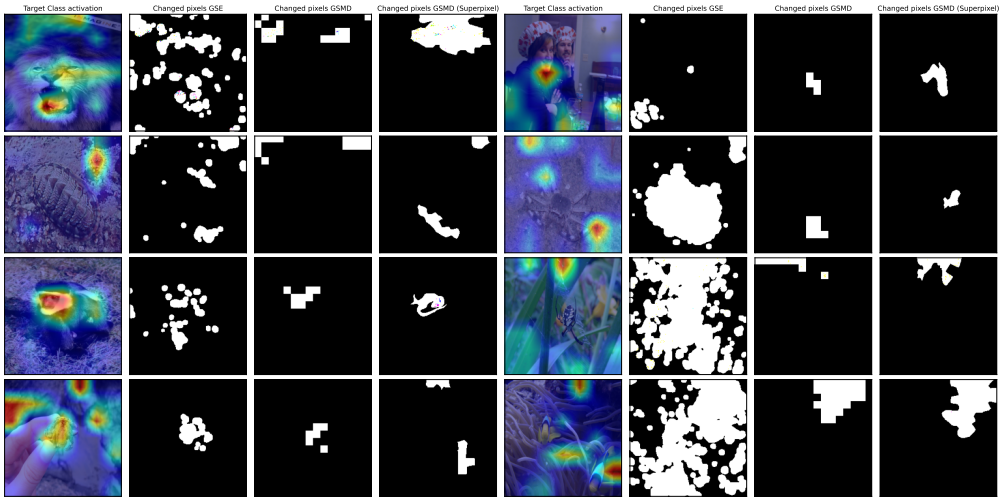


Figure 6: Visual explainability and qualitative analysis. This grid showcases 16 independent comparison sequences (with two side-by-side examples in each row) across a diverse range of ImageNet categories and spatial structures. The columns depict (from left to right): (1) target class activation heatmaps (Grad-CAM), (2) changed pixel masks by GSE (Sadiku et al., 2023), (3) changes by GSMD (fixed grid), (4) changes by GSMD (superpixel), with the pattern repeated on the right side of the figure. GSE produces massive, amorphous perturbation blobs that are scattered across the image, offering poor visual interpretability. Conversely, GSMD decouples group structure from the optimization to generate localized, spatially coherent, and geometrically structured perturbations that align with either a rigid grid or semantic superpixel boundaries. This visual precision directly simulates realistic localized threats, improving interpretability.

C.3 Visual Examples

To further demonstrate the spatial coherence and general applicability of our proposed Group Structured Mirror Descent (GSMD) framework across diverse categories, we provide additional qualitative visual results on 16 diverse examples from the ImageNet dataset in Figure 6.

D Omitted Proofs

Computing the exact Bregman projection directly across the complex intersection of group-norm and local box constraints is computationally prohibitive. To render this projection mathematically tractable, we systematically reduce the high-dimensional problem into a continuous resource allocation task. Fundamentally, this reduction requires solving a separable, strictly convex minimization problem with piecewise-differentiable objectives. Because the objective function is characterized by finitely many points of non-differentiability, we can isolate the optimal interval and solve the entire allocation problem exactly via a fast, sorting-based 1D root search. Assumption 2 formally establishes the structural properties and Algorithm 3 solves the allocation problem, which is proven in Theorem D.1.

Assumption 2. *For all $i \in \{1, \dots, d\}$, the function $\Phi_i : \mathbb{R} \to \mathbb{R}$ is strictly convex and attains its unique global minimum at $z_i \geq 0$. Furthermore, Φ_i is differentiable on $\mathbb{R} \setminus \{c_{1,i}, \dots, c_{j_i,i}\}$ with $\{c_{1,i}, \dots, c_{j_i,i}\} \subseteq \mathbb{R}_{\geq 0}$.*

Algorithm 3 Resource Allocation with Strictly Convex Cost

Require: $c \in \mathbb{R}^d_{\geq 0}$, $\Phi_1, \ldots, \Phi_d$ satisfying Assumption 2.

$C_i = \{\min\{c_i, z_i\}\} \cup \{c_{1,i}, \ldots, c_{j_i,i}\}$
$\mathcal{G} = \{0\} \cup (\bigcup_{j=1}^{k} \bigcup_{c \in C_j} \{-\Phi'_{j,+}(c), -\Phi'_{j,-}(c)\})$
Sort $\mathcal{G}$ in ascending order
for $g \in \mathcal{G}$ **do**
  if $\sum_{j=1}^{k} \max\left\{\min\{\nabla\Phi_j^*(-g), c_j\}, 0\right\} \leq D$ **then**
    $A = \{j \in \{1 \ldots k\} \mid g \notin \partial\Phi_j(\min\{c_i, z_i\}) \text{ for all } i \in I_j\}$
    $L = \{j \in \{1 \ldots k\} \mid \nabla\Phi_j^*(-g) \leq 0\}$
    $U = \{j \in \{1 \ldots k\} \mid \nabla\Phi_j^*(-g) \geq c_{i_j}\}$
    Define $\Psi : \mathbb{R} \to \mathbb{R}, \lambda \mapsto \sum_{j \in A \setminus U \setminus L} \nabla\Phi_j^*(-\lambda)$
    $\lambda = \Psi^{-1}(D - \sum_{j \notin A} \nabla\Phi_j^*(-g) - \sum_{j \in U} c_{i_j})$
    $x_i \leftarrow \max\left\{\min\{\nabla\Phi_j^*(-\lambda), c_j\}, 0\right\}$ for $i = 1, \ldots, d$
    return x
  end if
end for

Theorem D.1. *Let $\Phi_1, \ldots \Phi_d$ be the functions satisfying Assumption 2. Given $c_i \in \mathbb{R}^d_{\geq 0}$ and $D > 0$, Algorithm 3 returns the solution to the optimization problem*

$$\min_{x \in \mathcal{C}(x)} \quad \sum_{i=1}^{d} \Phi_i(x_i)$$

with $\mathcal{C}(x) := \{x \in \mathbb{R}^d : \sum_{i=1}^{d} x_i \leq D, 0 \leq x_i \leq c_i, i = 1, \ldots, d\}$.

Proof of Theorem D.1. By convexity, the KKT conditions are necessary and sufficient. Such that, using the Lagrange multipliers,

$$\mathcal{L}(x, \lambda, \nu, \mu) = \sum_{i=1}^{d} \Phi_i(x_i) - \nu^\top x - \mu^\top (c - x) - \lambda(D - \sum_{i=1}^{d} x_i)$$

for $x \in \mathbb{R}^d$, $\lambda \in \mathbb{R}_{\geq 0}$, $\nu \in \mathbb{R}^d_{\geq 0}$ and $\mu \in \mathbb{R}^d_{\geq 0}$, it follows from the optimality condition that

$$\nabla\Phi_i(x_i^*) - \nu_i^* + \mu_i^* + \lambda^* = 0$$

holds for some $\nabla\Phi_i(x_i^*) \in \partial\Phi_i(x_i^*)$ (Beck, 2017). Since Φ_i is strictly convex, Φ_i^* differentiable. Thus, we have

$$x_i^* = \nabla\Phi_i^*(\nu_i^* - \mu_i^* - \lambda^*).$$

Using complementary slackness and properties of ϕ_i, we have

$$0 = \nabla\Phi_i^*(-\lambda^* + \nu_i^*) \geq \nabla\Phi_i^*(-\lambda^*),$$

$$c_i = \nabla\Phi_i^*(-\lambda^* - \mu_i^*) \leq \nabla\Phi_i(-\lambda^*)$$

and

$$x_i^* = \nabla\Phi_i^*(-\lambda^*)$$

for $0 < x_i^* < c_i$. Since $\nabla\phi_i$ is invertible, we have

$$\sum_{i=1}^{d} \max\{\min\{\nabla\Phi_i^*(-\lambda^*), c_i\}, 0\}) = \sum_{i=1}^{d} x_i^*.$$

Let z_i be the global minimum of Φ_i. For each coordinate $i \in \{1, \ldots, d\}$, let C_i denote the set of all critical primal points, encompassing the boundaries and the non-differentiable kinks:

$$C_i = \{\min\{c_i, z_i\}, 0\} \cup \{c_{1,i}, \ldots, c_{j_i,i}\}.$$

We then define the global set of critical dual breakpoints $\mathcal{G}$ by collecting the left and right derivatives of Φ_i at all points in C_i:

$$\mathcal{G} = \bigcup_{i=1}^{d} \bigcup_{x \in C_i} \{ -\Phi'_{i,-}(x), -\Phi'_{i,+}(x) \}.$$

By sorting the unique elements of $\mathcal{K}$ in ascending order, we obtain a sequence $g_1 < g_2 < \cdots < g_m$ that partitions the dual space into distinct open intervals. We define the budget evaluation function $\Psi : \mathbb{R} \to \mathbb{R}$ as:

$$\Psi(\lambda) = \sum_{i=1}^{d} \max \{ \min\{\nabla \Phi_i^*(-\lambda), c_i\}, 0 \}.$$

We claim that the optimal λ^* must be in the interval $[g_1, g_m]$. Since we have $0 \in [g_1, g_m]$ for all i, it holds clearly

$$\Psi(g_1) \geq \sum_{i=1}^{d} \max \{ \min\{\nabla \Phi_i^*(\nabla \Phi_i(\min\{c_i, z_i\})), c_i\}, 0 \} \geq \sum_{i=1}^{d} \min\{z_i, c_i\} \geq \Psi(\lambda^*).$$

Similarly, we have $-\Phi'_{i,-}(0) \in [g_1, g_m]$ for all i, which implies

$$\Psi(g_m) \leq \sum_{i=1}^{d} \max \{ \min\{\nabla \Phi_i^*(\Phi'_{i,-}(0)), c_i\}, 0 \} = 0 \leq \Psi(\lambda^*).$$

The function is well-defined due to the strict convexity of Φ_i, if we find an exact match $\Psi(g_j) = D$, then the optimal solution is given by

$$x_i^* = \max\{\min\{\nabla \Phi_i^*(-g_j), c_i\}, 0\}.$$

Otherwise there must be a j such that $\Psi(g_j) > D > \Psi(g_{j+1})$ due to the monotonicity of Ψ. Next, we show that $\Psi(\lambda)$ is bijective in (g_j, g_{j+1}) for all j. Let $A_j \subseteq \{1, \ldots, d\}$ denote the active set of indices, defined as the coordinates where $\nabla \Phi_i^*$ is strictly increasing in (g_j, g_{j+1}). We claim that $\nabla \Phi_i^*$ is constant for all $i \notin A_j$. Let $i \notin A_j$ be arbitrary. Then there is an interval $[a, b] \in (g_j, g_{j+1})$ with $a < b$ such that $\nabla \Phi_i^*(-g) = c$ for all $g \in [a, b]$, which implies

$$gc - \Phi_i(c) = \max_x gx - \Phi_i(x) >= gx - \Phi_i(x)$$

for all x and g, i.e. $[a, b] \subseteq \partial \Phi_i(c)$. Thus, Φ_i is not differentiable at c and there is some $\nabla \Phi_i(c) \in (g_j, g_{j+1})$. Now, if $\nabla^* \Phi_i$ is not constant in (g_j, g_{j+1}), either $\Phi'_{i,-}(c)$ or $\Phi'_{i,+}(c)$ is in (g_j, g_{j+1}), which is impossible due to the construction of $\mathcal{G}$. For all inactive indices $i \notin A_j$, the clipped conjugate gradient is locked at a constant value $c_{k_j,j}$ throughout the entire interval. Consequently, for any $\lambda \in (g_j, g_{j+1})$, the function $\Psi(\lambda)$ simplifies exactly to a strictly monotonic component and a pre-computable constant:

$$\Psi(\lambda) = \sum_{i \in A_j} \max \{ \min\{\nabla \Phi_i^*(-\lambda), c_i\}, 0 \} + \sum_{i \notin A_j} c_{k_j,j}.$$

Thus the optimal λ^* can be found by solving

$$\sum_{i \in A_j} \max \{ \min\{\nabla \Phi_i^*(-\lambda), c_i\}, 0 \} = D - \sum_{i \notin A_j} c_{k_j,j}.$$

Finally, since $\nabla \Phi_i(c_i)$ and $\nabla \Phi_i(0)$ are also in $\mathcal{G}$ for all i the coordinates, where constraints are active within the interval (g_j, g_{j+1}), are also fixed. Specifically, we define L_j and U_j as the sets of indices locked at the lower bound (0) and upper bound (c_i), respectively:

$$L_j = \{i \in \{1 \ldots d\} \mid \nabla \Phi_i^*(-\lambda) \leq 0, \ \forall \lambda \in (g_j, g_{j+1})\}$$
$$U_j = \{i \in \{1 \ldots d\} \mid \nabla \Phi_i^*(-\lambda) \geq c_i, \ \forall \lambda \in (g_j, g_{j+1})\}.$$

Thus, we obtain

$$\Psi_j(\lambda^*) = \sum_{i \in A_j \setminus U_j \setminus L_j} \nabla \Phi_i^*(-\lambda^*) = D - \sum_{i \notin A_j} c_{k_i,i} - \sum_{i \in U_j} c_i.$$

Since Ψ_j is strictly decreasing, i.e. bijective, we find the optimal

$$\lambda^* = \Psi_j^{-1}(D - \sum_{i \notin A_j} c_{k_i,i} - \sum_{i \in U_j} c_i).$$

Thus, we obtain the optimal solution by applying Algorithm 3. □

E Application to Bregman Projection

Proof of Lemma 3.2. For simplicity, we assume w.l.o.g. $z_i \geq 0$ and $x_i^* \geq 0$. Let $j \in \{1, \ldots, k\}$ and $i \in I_j$ be arbitrary. Since the function is strictly convex and symmetric, the function $x \mapsto (\phi(x) - \phi'(z_i)x)$ is strictly decreasing on $[0, z_i]$ and strictly increasing on $[z_i, \infty)$. Thus, we have $x_i^* \leq \min\{z_i, c_i\}$ and the claim holds clearly for $i = i_j$. Otherwise, we consider three cases.

For the case $x_{i_j}^* \geq \min\{z_i, c_i\}$ **and** $x_i^* < \min\{z_i, c_i\}$, we can increase x_i^* to decrease the objective without violating the feasibility.

For the case $x_{i_j}^* \leq \min\{z_i, c_i\}$ **and** $x_i^* < x_{i_j}^* \leq \min\{z_i, c_i\}$, setting $x_i^* = x_{i_j}^*$ can decrease the objective without violating the feasibility.

For the case $x_{i_j}^* \leq \min\{z_i, c_i\}$ **and** $x_i^* > x_{i_j}^*$, we have

$$x_{i_j}^* < x_i^* \leq \min\{z_i, c_i\} \leq \max\{\min\{z_l, c_l\} | l \in I_j\} = \min\{z_{i_j}, c_{i_j}\},$$

and we can increase $x_{i_j}^*$ to decrease the objective without violating the feasibility. The claim is proved by the conclusion of the three cases. □

Proof of Proposition 3.1. Since ϕ is symmetric, the sign of a non-zero coordinate i of an optimal solution must be the same as the sign of z_i. Thus, we can assume w.l.o.g. that $x_i \geq 0$ and $z_i \geq 0$ for all i. The problem can be reduced to

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \sum_{i=1}^{d} (\phi(x_i) - \phi'(y_i)x_i) \\ \text{s.t.} \quad & \sum_{j=1}^{k} \max_{i \in I_j} x_i \leq D \\ & x_i \geq 0 \text{ for all } i = 1, \ldots, d \\ & x_i \leq c_i \text{ for all } i = 1, \ldots, d. \end{aligned} \tag{11}$$

The feasible set is clearly convex. It is an intersection of a box and the inverse image of the closed set $[0, D]$ and the continuous function $x \mapsto \sum_{j=1}^{k} \max_{i \in I_j} x_i$. Being a closed subset of a compact set, it is therefore compact. Since the objective function is continuous and strictly convex over the compact feasible set, it attains a unique global minimum. Let $i_1, \ldots, i_k$ be the indices such that

$$i_j \in \arg\max\{\min\{z_l, c_l\} | l \in I_j\}.$$

for $j = 1, \ldots, k$.

Thus, we can tighten the feasible set of equation 11, and obtain the optimal solution by solving

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & \sum_{i=1}^{d} (\phi(x_i) - \phi'(z_i)x_i) \\ \text{s.t.} \quad & \sum_{j=1}^{k} x_{i_j} \leq D \\ & x_i \leq x_{i_j} \text{ for all } i \in I_j, j = 1, \ldots, k \\ & x_i \geq 0 \text{ for all } i = 1, \ldots, d \\ & x_i \leq c_i \text{ for all } i = 1, \ldots, d. \end{aligned} \tag{12}$$

Define

$$\begin{aligned} \Phi_j : \mathbb{R} \to \mathbb{R}, x \mapsto & \phi(x) - \phi'(z_{i_j})x \\ & + \sum_{i \in I_j, i \neq i_j} (\phi(\min\{z_i, c_i, x\}) - \phi'(z_i) \min\{z_i, c_i, x\}). \end{aligned}$$

We claim that, the optimization problem equation 12 can be reduced to

$$\begin{aligned} \min_{x\in\mathbb{R}^k} \quad & \sum_{j=1}^{k} \Phi_j(x_j) \\ \text{s.t.} \quad & \sum_{j=1}^{k} x_j \le D \\ & x_j \ge 0 \text{ for all } i = 1, \dots, k \\ & x_j \le c_j \text{ for all } i = 1, \dots, k. \end{aligned} \tag{13}$$

Let $\tilde{x}$ be the optimal solution of equation 13. We set $x^*_{i_j} = \tilde{x}_j$ and $x^*_i = \min\{\tilde{x}_j, c_i, z_i\}$ for all $i \in I_j, i \neq i_j$. Then, for all $x \in \mathbb{R}^d$ satisfying constraints of equation 12, we obtain

$$\begin{aligned} & \sum_{i=1}^{d} (\phi(x_i) - \phi'(z_i)x_i) \\ \ge & \sum_{j=1}^{k} (\phi(x_{i_j}) - \phi'(z_{i_j})x_{i_j} + \sum_{i\in I_j, i\neq i_j} (\phi(\min\{c_i, z_i, x_{i_j}\}) - \phi'(z_i)\min\{c_i, z_i, x_{i_j}\})) \\ = & \sum_{j=1}^{k} \Phi_j(x_{i_j}) \\ \ge & \sum_{j=1}^{k} \Phi_j(\tilde{x}_k) \\ = & \sum_{j=1}^{k} \sum_{i\in I_j} (\phi(x^*_i) - \phi'(z_i)x^*_i), \end{aligned}$$

where the first inequality holds because we can always decrease the objective by increasing x_i to $\min\{c_i, z_i, x_{i_j}\}$ for all $x_i < x_{i_j}, i \in I_j$ without violating the constraints. Since $x^* \in \mathbb{R}^d$ is a feasible solution of equation 12, it is the optimal solution of equation 12. Finally, Φ_j is strictly convex and differentiable at $\mathbb{R} \setminus \{c_i | i \in I_j, i \neq i_j\}$, the problem can be solved by applying Algorithm 3.

□

F IMPLEMENTATION

In this section, we provide the concrete mathematical derivations required to execute the exact Bregman projection (Algorithm 1) in practice. While the theoretical framework in Section D establishes the reduction of the projection to a 1D root-finding problem, an efficient algorithmic implementation requires explicitly calculating the breakpoints of the piecewise-differentiable cost function and evaluating their corresponding subgradients. Furthermore, to instantiate the Mirror Descent update, we must specify the potential function. As outlined in Section 3, we utilize a smoothed entropic generating function to natively navigate the non-Euclidean geometry of the clipped cross-polytope constraint space. Below, we derive the explicit left and right derivatives necessary to construct the sorted dual space $\mathcal{G}$, formulate the subgradients of the conjugate function, and finally derive the closed-form analytical solution for the resource allocation root (Ψ^{-1}) utilized in our structurally relaxed formulation.

Define

$$A^-_j(x) = \{i \in I_j \setminus \{i_j\} | x \le \min\{|z_i|, c_i\}\}$$

and

$$A^+_j(x) = \{i \in I_j \setminus \{i_j\} | x < \min\{|z_i|, c_i\}\}.$$

We have

$$\Phi'_{j,-}(x) = \phi'(x) - \phi'(|z_{i_j}|) + \sum_{i\in A^-_j(x)} (\phi'(x) - \phi'(|z_i|))$$

and

$$\Phi'_{j,+}(x) = \phi'(x) - \phi'(|z_{i_j}|) + \sum_{i \in A_j^+(x)} (\phi'(x) - \phi'(|z_i|)).$$

Thus, $\partial\Phi_j(x)$ is not unique only for $x \in \{\min\{c_i, z_i\} \mid i \in I_j\}$. Let $v_i = \min\{|z_i|, c_i\}$, and ρ be a permutation such that $v_{\rho(i)} \leq v_{\rho(i+1)}$ for all i. Given $g \in \mathcal{G}$ and j, we construct $\nabla\Phi_j^*(-g)$ as follows.

For $j \notin A$, there must be some $i \in I_j$ such that $-g \in \partial\Phi_j(v_i)$, i.e. $\nabla\Phi_j^*(-g) = v_i$. For $j \in A$, we define $A_j = \{i \in I_j \mid \Phi'_{j,-}(v_i) > -g\}$. The subgradient of the conjugate function is given by

$$\phi'(\nabla\Phi_j^*(-\lambda)) = \frac{1}{|A_j|}(\sum_{i \in A_j} \phi'(|z_i|) - \lambda).$$

For

$$\phi : \mathbb{R} \to \mathbb{R}, \quad x \mapsto (|x| + \beta)\ln\left(\frac{|x|}{\beta} + 1\right) - |x|,$$

we have

$$\begin{aligned}\nabla\Phi_j^*(-\lambda) =& \beta \exp(\frac{1}{|A_j|}(\sum_{i \in A_j} \log(\frac{|z_i|}{\beta} + 1) - \lambda)) - \beta \\ =& (\Pi_{i \in A_j}(|z_i|) + \beta)^{\frac{1}{|A_j|}} \exp(-\frac{\lambda}{|A_j|}) - \beta\end{aligned}$$

Thus,

$$\Psi(-\lambda) = \sum_{j \in A \setminus U \setminus L} ((\Pi_{i \in A_j}(|z_i|) + \beta)^{\frac{1}{|A_j|}} \exp(-\frac{\lambda}{|A_j|}) - \beta).$$

In general, there is no closed-form expression of Ψ^{-1}. However, if there is one coordinate in each group, we have

$$\Psi(-\lambda) = \sum_{i \notin U \cup L} ((|z_i| + \beta)\exp(-\lambda) - \beta).$$

The inverse in Line 12 of Algorithm 1 is given by

$$\lambda = \log \frac{\sum_{i \notin U \cup L}(z_i + \beta)}{D - \sum_{i \in U} c_i + \sum_{i \notin U \cup L} \beta}.$$

Adaptive Group Budget In the following, we outline a practical adaptive perturbation budget based on the logits of the target class of the perturbed input. Suppose the classifier $f : \mathcal{X} \to \mathbb{R}^K$ for a fixed input $x \in \mathcal{X}$ with target class $y \neq y^*$, we write $y^* = \arg\max_c f_c(x)$ for the clean prediction. Consider, the perturbation δ satisfies the box-constraint and budget constraint in (2) denoted as $C(D)$, for a given δ and target y, we define the target margin as

$$m(\delta) = f_y(x + \delta) - \max_{c \neq y} f_c(y + \delta).$$

A targeted attack is successful iff. $m(\delta) \geq 0$. We write $\Psi(D) := \sup_{\delta \in C(D)} m(\delta)$ for the achievable margin for a given perturbation budget D, where the minimal budget for a targeted success is

$$D^* := \inf\{D \geq 0 : \Psi(D) \geq 0\}.$$

Since $\Psi(D)$ is unknown in general, we suggest an adaptive estimate of the perturbation budget D_t based on the observed margin at iteration t. Let δ_t denote the current perturbation at satisfying $\sum_{j=1}^k \max_{I_j} |\delta_i| \leq D_t$, then we use the following multiplicative update

$$D_{t+1} = \Pi_{[D_{\min}, D_{\max}]}(D_t \gamma_t)$$

with

$$\gamma_t := \begin{cases} 1 + \rho_t, & m_t(D_t) < 0, \\ 1 - \rho_t, & m_t(D_t) \geq 0 \end{cases}$$

for $\rho_t \in (0, 1)$, with $\rho_t \to 0$ for $t \to \infty$, $\sum_{t=1}^\infty \rho_t = \infty$ and $\sum_{t=1}^\infty \rho_t^2 < \infty$.