SAFE: Finding Sparse and Flat Minima to Improve Pruning

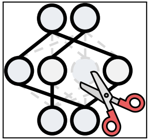
ICML 2025 Spotlight Poster

Dongyeop Lee   Kwanhee Lee   Jinseok Chung   Namhoon Lee

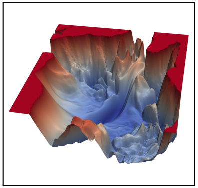
June 2025

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Performance degradation during pruning may be due to loss sharpness

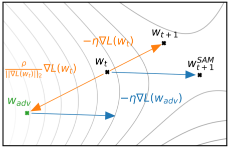


- Pruning during training has proven effective in achieving good sparse network (Hoefler et al. 2021)
- Still, they often lead to diminished model trainability and generalization performance



- Recent studies analyzed these through the lens of optimization geometry, hinting at the sharpness of the loss as its cause (Keskar et al. 2017; Lee et al. 2021)

Idea: explicitly penalize sharpness while pruning



- To recover this, we attend to sharpness minimization (Foret et al. 2021)
- The aim is to induce flat minima, which is shown to improve generalization effectively
- We propose *Sparsification via ADMM with Flatness Enforcement* or SAFE: a principled approach to enforcing flatness simultaneously with sparsity

Problem formulation for finding sparse and flat minima

We first formulate this as a sharpness-aware sparsity-constrained optimization problem:

$$\min_{\|x\|_0 \leq d} \max_{\|\epsilon\|_2 \leq \rho} f(x + \epsilon),$$

where goal is to find a sparse solution $x^\star$ with atmost d non-zero elements that minimizes the objective function in the whole ϵ-neighborhood, *i.e.*, seek flat minima.

Augmented Lagrangian based approach

To solve this, we form the augmented Lagrangian dual problem of the following:

$$\max_{u}, \min_{x,z} \left[\mathcal{L}(x,z,u) := \max_{\|\epsilon\|_2 \leq \rho} f(x+\epsilon) + I_{\|\cdot\|_0 \leq d}(z) - \frac{\lambda}{2}\|u\|_2^2 + \frac{\lambda}{2}\|x - z + u\|_2^2 \right],$$

where we separate the sparsity-constraint satisfaction using variable z so that it can be handled more easily.

Alternating Direction Method of Multipliers

Applying dual ascent, where we minimize x and z in an alternating fashion, gives us the following ADMM iterate:

$$
\begin{aligned}
x_{k+1} &= \underset{x}{\operatorname{argmin}} \max_{\|\epsilon\|_2 \leq \rho} f(x+\epsilon) + \frac{\lambda}{2}\|x - z_k + u_k\|_2^2 \\
z_{k+1} &= \underset{z}{\operatorname{argmin}} I_{\|\cdot\|_0 \leq d}(z) + \frac{\lambda}{2}\|x - z + u\|_2^2 \\
u_{k+1} &= u_k + x_{k+1} - z_{k+1},
\end{aligned}
$$

x-minimization: iterative minimization while enforcing flatness

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \max_{\|\epsilon\|_2 \leq \rho} f(x+\epsilon) + \frac{\lambda}{2}\|x - z_k + u_k\|_2^2$$

We solve this iteratively using *Sharpness-aware minimization (SAM)* (Foret et al. 2021), where we approximately solve for ϵ through first-order Taylor approximation:

$$\epsilon^\star(x) \approx \underset{\|\epsilon\|_2 \leq \rho}{\operatorname{argmax}}\ f(x) + \epsilon^\top \nabla f(x) = \rho \frac{\nabla f(x)}{\|\nabla f(x)\|_2}.$$

Applying this back to the objective and applying gradient descent gives us the following iteration for x-minimization

$$x_k^{(t+1)} = x_k^{(t)} - \eta^{(t)} \left[\nabla f\left(x_k^{(t)} + \rho \frac{\nabla f(x_k^{(t)})}{\|\nabla f(x_k^{(t)})\|_2} \right) + \lambda (x_k^{(t)} - z_k + u_k) \right],$$

z-minimization: Euclidean projection onto sparsity constraint

z-minimization corresponds to projecting $x_{k+1} + u_k$ onto the sparsity constraint in terms of Euclidean distance

$$\begin{aligned} z_{k+1} &= \underset{z}{\text{argmin}} I_{\|\cdot\|_0 \leq d}(z) + \frac{\lambda}{2}\|x_{k+1} - z + u_k\|_2^2 \\ &= \text{proj}_{\|\cdot\|_0 \leq d}(x_{k+1} + u_k). \end{aligned}$$

This leads to the classic hard thresholding operator, where we zero out except d elements with the largest magnitude

$\textsc{Safe}^{+}$: Generalized projection

However, this magnitude-based projection often yields subpar performance in practice.

To improve this, we introduce a generalized distance $\frac{1}{2}\|\cdot\|_{\mathbf{P}}^2$ with diagonal positive definite matrix $\mathbf{P}$:

$$
\begin{aligned}
z_{k+1} &= \mathrm{proj}_{\|\cdot\|_0 \leq d}^{\mathbf{P}}(x_{k+1} + u_k) \\
&:= \underset{\|z\|_0 \leq d}{\arg\min} \frac{1}{2}\|z - (x_{k+1} + u_k)\|_{\mathbf{P}}^2 \\
&= \underset{\|z\|_0 \leq d}{\arg\min} \frac{1}{2}(z - (x_{k+1} + u_k))^\top \mathbf{P}(z - (x_{k+1} + u_k)).
\end{aligned}
$$

SAFE$^+$: Generalized projection (cont.)

Criteria	$\mathbf{P}$
Magnitude	$\mathbf{I}$
OBD	$\mathrm{diag}(H)$
SNIP	$\mathrm{diag}(\nabla f \nabla f^\top)$
Wanda	$\mathrm{diag}(\mathbf{A}^\top \mathbf{A})$

- This generalized projection framework allows us to employ various saliency scores within the projection step
- Here we use this primarily for LLM pruning, though it is generally applicable to other domains

Final algorithm: Safe and Safe$^+$

Algorithm Safe and Safe$^+$ algorithms

Require: Target parameter count d, total train iteration T, dual-update interval K, learning rate $\eta^{(t)}$, perturbation radius ρ, penalty parameter λ, importance matrix $\mathbf{P}$.

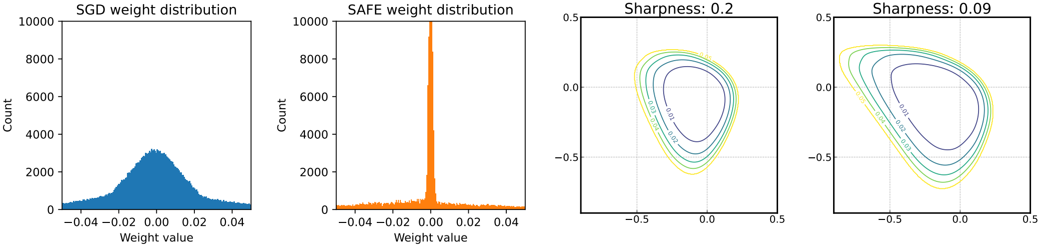
```
Initialize $x^{(0)}$
$u = \mathbf{0}$
for $t$ in $T$ do
    if $t \bmod K = 0$ then
        if Safe then
            $z = \mathrm{proj}_{\|\cdot\|_0 \le d}(x^{(t+1)} + u)$
        else if Safe$^+$ then
            $z = \mathrm{proj}^{\mathbf{P}}_{\|\cdot\|_0 \le d}(x^{(t+1)} + u)$
        end if
        $u = u + x^{(t+1)} - z$
    end if
    $x^{(t+1/2)} = x^{(t)} - \eta^{(t)} \nabla f\left(x^{(t)} + \rho \cdot \frac{\nabla f(x^{(t)})}{\|\nabla f(x^{(t)})\|_2}\right)$
    $x^{(t+1)} = x^{(t+1/2)} - \eta^{(t)} \lambda (x^{(t)} - z + u)$
end for
return $\mathrm{proj}_{\|\cdot\|_0 \le d}(x^{(T)})$ =0
```

- Registers sparse point closest to the current x to z every few steps
- Penalizes x iterate to move slightly closer to z during flatness-inducing minimization.
- This gradually moves x towards sparsity during flatness induction without sudden changes, yielding a sparse and flat minima.

Convergence analysis

Corollary 1. (Convergence of SAFE) Suppose that f is smooth and weakly convex. Assume further that δ is chosen large enough so that $\delta^{-1}\beta^2 - (\delta - \mu)/2 < 0$. Let $(\bar{x}, \bar{z}, \bar{u})$ be a limit point of SAFE algorithm. Then $\bar{x}$ is a δ-stationary point of the sparsity-constrained optimization problem.

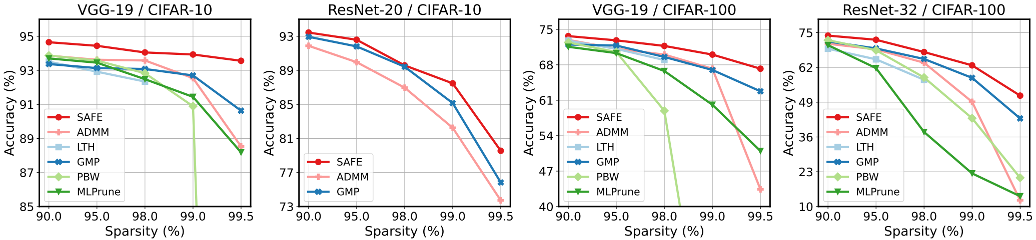
Result: Safe finds sparse and flat solutions



(a) Dense training (b) Safe (c) ADMM (d) Safe

(a-b) Weight distributions of densely-trained model and model trained with Safe, and (c-d) loss landscape and maximum Hessian eigenvalue of minima found by ADMM and Safe. Safe yields sparse and flat solutions.

Result: Improved generalization performance in Image classification



SAFE outperforms other baselines in various image classification tasks

Result: Improved generalization performance in LLM post-training pruning

		LLaMa-2				LLaMa-3	
		7B		13B		8B	
Sparsity	Method	Wikitext	C4	Wikitext	C4	Wikitext	C4
0%	Dense	5.47	7.26	4.88	6.72	6.23	9.53
50%	Magnitude	16.03	21.33	6.82	9.37	134.20	273.3
	SparseGPT	6.99$_{\pm 0.03}$	9.20$_{\pm 0.03}$	6.06$_{\pm 0.03}$	8.20$_{\pm 0.01}$	9.36$_{\pm 0.11}$	13.96$_{\pm 0.02}$
	Wanda	6.92$_{\pm 0.01}$	9.23$_{\pm 0.00}$	5.98$_{\pm 0.01}$	8.28$_{\pm 0.01}$	9.71$_{\pm 0.03}$	14.88$_{\pm 0.04}$
	ALPS	6.87$_{\pm 0.01}$	8.98$_{\pm 0.00}$	5.96$_{\pm 0.02}$	8.09$_{\pm 0.04}$	9.05$_{\pm 0.12}$	13.40$_{\pm 0.06}$
	SAFE	6.78$_{\pm 0.01}$	8.93$_{\pm 0.00}$	5.76$_{\pm 0.01}$	7.85$_{\pm 0.02}$	9.59$_{\pm 0.06}$	14.60$_{\pm 0.04}$
	SAFE^{+}	**6.56**$_{\pm 0.01}$	**8.71**$_{\pm 0.00}$	**5.67**$_{\pm 0.01}$	**7.74**$_{\pm 0.01}$	**8.62**$_{\pm 0.06}$	**13.26**$_{\pm 0.06}$
60%	Magnitude	1864	2043	11.81	14.62	5335	7438
	SparseGPT	10.19$_{\pm 0.08}$	12.86$_{\pm 0.05}$	8.31$_{\pm 0.09}$	10.85$_{\pm 0.09}$	15.46$_{\pm 0.40}$	21.25$_{\pm 0.18}$
	Wanda	10.75$_{\pm 0.07}$	13.87$_{\pm 0.01}$	8.43$_{\pm 0.07}$	11.55$_{\pm 0.01}$	22.06$_{\pm 0.19}$	32.28$_{\pm 0.37}$
	ALPS	9.55$_{\pm 0.00}$	11.24$_{\pm 0.03}$	7.54$_{\pm 0.03}$	9.87$_{\pm 0.05}$	14.03$_{\pm 0.35}$	18.72$_{\pm 0.15}$
	SAFE	9.20$_{\pm 0.04}$	11.51$_{\pm 0.04}$	7.18$_{\pm 0.03}$	9.59$_{\pm 0.03}$	15.90$_{\pm 0.25}$	22.26$_{\pm 0.16}$
	SAFE^{+}	**8.30**$_{\pm 0.06}$	**10.59**$_{\pm 0.00}$	**6.78**$_{\pm 0.04}$	**9.02**$_{\pm 0.15}$	**12.18**$_{\pm 0.22}$	**17.30**$_{\pm 0.02}$
4:8	Magnitude	15.91	31.61	7.32	9.96	212.5	336.3
	SparseGPT	8.42$_{\pm 0.05}$	10.73$_{\pm 0.03}$	7.02$_{\pm 0.06}$	9.33$_{\pm 0.04}$	12.16$_{\pm 0.20}$	17.36$_{\pm 0.06}$
	Wanda	8.64$_{\pm 0.03}$	11.35$_{\pm 0.01}$	7.01$_{\pm 0.02}$	9.70$_{\pm 0.03}$	13.84$_{\pm 0.04}$	21.14$_{\pm 0.06}$
	ALPS	8.11$_{\pm 0.09}$	10.21$_{\pm 0.04}$	6.81$_{\pm 0.07}$	9.33$_{\pm 0.04}$	11.38$_{\pm 0.17}$	16.10$_{\pm 0.10}$
	SAFE	8.21$_{\pm 0.01}$	10.61$_{\pm 0.04}$	6.60$_{\pm 0.02}$	8.95$_{\pm 0.02}$	12.15$_{\pm 0.14}$	17.90$_{\pm 0.15}$
	SAFE^{+}	**7.59**$_{\pm 0.03}$	**9.88**$_{\pm 0.01}$	**6.37**$_{\pm 0.03}$	**8.61**$_{\pm 0.01}$	**10.51**$_{\pm 0.13}$	**15.67**$_{\pm 0.02}$
2:4	Magnitude	37.77	74.70	8.88	11.72	792.8	2245
	SparseGPT	11.00$_{\pm 0.20}$	13.54$_{\pm 0.03}$	8.78$_{\pm 0.09}$	11.26$_{\pm 0.11}$	15.87$_{\pm 0.32}$	22.45$_{\pm 0.12}$
	Wanda	12.17$_{\pm 0.02}$	15.60$_{\pm 0.11}$	9.01$_{\pm 0.04}$	12.40$_{\pm 0.01}$	23.03$_{\pm 0.38}$	34.91$_{\pm 0.31}$
	ALPS	9.99$_{\pm 0.19}$	12.04$_{\pm 0.04}$	8.16$_{\pm 0.17}$	10.35$_{\pm 0.18}$	14.53$_{\pm 0.33}$	19.74$_{\pm 0.18}$
	SAFE	10.53$_{\pm 0.13}$	13.20$_{\pm 0.07}$	7.64$_{\pm 0.05}$	10.10$_{\pm 0.01}$	17.49$_{\pm 0.27}$	24.45$_{\pm 0.13}$
	SAFE^{+}	**8.96**$_{\pm 0.07}$	**11.34**$_{\pm 0.03}$	**7.20**$_{\pm 0.04}$	**9.52**$_{\pm 0.01}$	**13.39**$_{\pm 0.23}$	**19.03**$_{\pm 0.01}$

- SAFE achieves competitive performance, while SAFE^{+} outperforms baselines across all settings.

Results: Robustness under label noise

Sparsity	Method	Noise ratio 25%	50%	75%
70%	ADMM	$77.00_{\pm 0.91}$	$59.18_{\pm 0.55}$	$32.62_{\pm 0.89}$
	SAFE	**90.58**$_{\pm 0.30}$	**86.51**$_{\pm 0.16}$	**67.01**$_{\pm 0.54}$
80%	ADMM	$76.18_{\pm 0.56}$	$62.67_{\pm 0.38}$	$32.86_{\pm 1.12}$
	SAFE	**91.25**$_{\pm 0.12}$	**86.55**$_{\pm 0.07}$	**66.49**$_{\pm 0.56}$
90%	ADMM	$79.40_{\pm 0.12}$	$66.64_{\pm 0.13}$	$36.84_{\pm 0.94}$
	SAFE	**90.68**$_{\pm 0.21}$	**86.49**$_{\pm 0.06}$	**64.72**$_{\pm 0.61}$
95%	ADMM	$77.71_{\pm 0.52}$	$67.10_{\pm 1.37}$	$39.68_{\pm 1.44}$
	SAFE	**89.86**$_{\pm 0.11}$	**85.18**$_{\pm 0.15}$	**64.25**$_{\pm 0.36}$

- Noisy label training. Validation accuracy is measured for sparse models trained with ADMM and SAFE under various levels of label noise and sparsity.
- SAFE is much more robust to label noise.

Results: Robustness to common image corruptions and adversarial attacks

Sparsity	Method	Common corruption (avg.)		Adversarial	
		intensity=3	intensity=5	l_∞-PGD	l_2-PGD
90%	ADMM	$70.06_{\pm 0.03}$	$52.01_{\pm 0.38}$	$49.81_{\pm 1.02}$	$49.71_{\pm 1.06}$
	SAFE	**73.98**$_{\pm 0.09}$	**55.11**$_{\pm 0.27}$	**56.43**$_{\pm 1.03}$	**56.36**$_{\pm 1.11}$
95%	ADMM	$68.87_{\pm 0.25}$	$50.56_{\pm 0.07}$	$49.84_{\pm 1.78}$	$49.68_{\pm 1.79}$
	SAFE	**72.92**$_{\pm 0.41}$	**54.86**$_{\pm 0.51}$	**51.40**$_{\pm 0.89}$	**51.36**$_{\pm 0.94}$
98%	ADMM	$65.46_{\pm 0.24}$	$48.65_{\pm 0.04}$	$43.33_{\pm 1.59}$	**43.42**$_{\pm 1.60}$
	SAFE	**68.20**$_{\pm 0.47}$	**49.96**$_{\pm 0.83}$	**43.34**$_{\pm 0.90}$	$43.41_{\pm 1.03}$
99%	ADMM	$59.21_{\pm 0.47}$	$43.81_{\pm 0.44}$	$30.29_{\pm 0.64}$	$30.32_{\pm 0.58}$
	SAFE	**66.02**$_{\pm 0.56}$	**49.34**$_{\pm 1.03}$	**43.70**$_{\pm 1.28}$	**32.70**$_{\pm 1.28}$
99.5%	ADMM	$55.72_{\pm 0.44}$	$41.55_{\pm 0.78}$	$23.25_{\pm 1.92}$	$23.25_{\pm 1.85}$
	SAFE	**56.58**$_{\pm 0.36}$	**42.27**$_{\pm 0.63}$	**29.48**$_{\pm 0.68}$	**29.45**$_{\pm 0.74}$

- Evaluation on corrupted data. CIFAR-10C is used for common corruptions, and l_∞ and l_2 PGD attacks are used to generate adversarial corruption on the validation set of CIFAR-10.
- SAFE improves robustness over naturally and adversarially corrupted images.

Results: Comparison with other SAM-based pruners

Method	Sparsity 95%	98%	99%	99.5%
IMP+SAM_{linear}	$80.30_{\pm 0.12}$	$36.03_{\pm 4.19}$	$18.30_{\pm 2.80}$	$13.80_{\pm 0.52}$
IMP+SAM_{cubic}	$92.50_{\pm 0.05}$	$89.24_{\pm 0.06}$	$83.74_{\pm 0.14}$	$73.73_{\pm 0.30}$
CrAM	$90.18_{\pm 1.80}$	$69.53_{\pm 12.36}$	$45.17_{\pm 20.86}$	$10.00_{\pm 0.00}$
$CrAM^{+}$	**93.62**$_{\pm 0.06}$	**91.75**$_{\pm 0.41}$	$\underline{88.82}_{\pm 0.18}$	$\underline{81.30}_{\pm 0.56}$
SAFE	$\underline{92.59}_{\pm 0.09}$	$89.58_{\pm 0.10}$	$87.47_{\pm 0.07}$	$79.55_{\pm 0.13}$
$SAFE_{+SG}$	$92.40_{\pm 0.06}$	$\underline{90.09}_{\pm 0.13}$	**89.13**$_{\pm 0.06}$	**85.85**$_{\pm 0.09}$

- Comparison with IMP+SAM, CrAM, and $CrAM^{+}$ on ResNet-20/CIFAR-10.
- $SAFE_{+SG}$, which extends SAFE using a similar technique as $CrAM^{+}$, outperforms most baselines at moderate sparsity and all baselines at extreme sparsity.

Conclusion

- We propose SAFE and SAFE^{+}: an optimization-based approach to find flat and sparse minima to improve pruning
- It improves performance across standard image classification and language model post-training pruning tasks
- SAFE also shows robust performance under label noise training, common image corruptions, and adversarial attacks
- Finally, compared to other SAM-based pruners, it shows strong performance even at extreme sparsities unlike other baselines.

Pytorch

Jax

References I

Foret, Pierre et al. (2021). "Sharpness-aware minimization for efficiently improving generalization". In: *ICLR*.

Hoefler, Torsten et al. (2021). "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks". In: *JMLR*.

Keskar, Nitish Shirish et al. (2017). "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima". In: *ICLR*.

Lee, Namhoon et al. (2021). "Understanding the effects of data parallelism and sparsity on neural network training". In: *ICLR*.