F-WANDA: Fisher-Reweighted Post-Training Pruning for Sustainable Deployment of Large Language Models

Anonymous ACL submission

Abstract

One-shot post-training pruning is the most energy-frugal compression strategy for large language models (LLMs), yet existing approaches trade either quality (WANDA) or compute cost (SPARSEGPT). We introduce F-WANDA, a drop-in modification of WANDA that reallocates the per-row keep budget across output neurons in proportion to the empirical Fisher information of the pre-activation. The Fisher signal is collected in a single additional backward pass over the same calibration corpus WANDA already uses; no weights are updated. On LLAMA-2-7B at 50 % unstructured sparsity, F-WANDA attains WikiText-2 perplexity of **6.85**, matches WANDA fluency, and improves 5-shot MMLU by **+1.6 pp** over WANDA and **+1.1 pp** over SPARSEGPT, while incurring only one-third of SPARSEGPT's pruning wall-clock and energy. The headline trade-off is achieved without extra calibration data or fine-tuning, placing F-WANDA on the Pareto frontier of quality versus pruning cost for sustainable LLM compression.

1 Introduction

Deploying state-of-the-art large language models is dominated by inference-side compute and energy: a single LLAMA-2-13B forward pass on an H100 GPU costs roughly *40 J/token*, and aggregate inference energy now exceeds training energy for many production systems (Patterson et al., 2021). One-shot post-training pruning is the cheapest compression family that touches this cost directly: it requires neither retraining nor labelled data, only a small calibration corpus to estimate weight importance (Frantar and Alistarh, 2023; Sun et al., 2024).

Two methods dominate the recent literature. WANDA ranks weights by the product of their magnitude and the input-activation norm; it requires only one forward pass and is therefore extremely cheap, but spends its sparsity budget *uniformly* across output neurons. SPARSEGPT performs an Optimal-Brain-Surgeon (Hassibi and Stork, 1993) style column sweep with per-block Hessian inversion and explicit weight updates; quality is higher but the procedure is an order of magnitude more expensive and considerably more delicate to implement.

We argue that WANDA's uniform per-row budget is the bottleneck for knowledge-intensive tasks: output neurons whose pre-activations consistently drive the loss carry stored facts disproportionately, yet WANDA prunes them at the same rate as "confident" neurons whose gradient is near zero. We address this with a single additional backward pass that yields a per-neuron empirical Fisher scalar, used as the reallocation weight for the existing WANDA score. We call the resulting method F-WANDA.

Contributions.

- A drop-in modification of WANDA that reallocates the per-row sparsity budget by the empirical Fisher of each output neuron, with no weight updates and no extra calibration data.
- An efficient mask realisation via a water-filling, largest-remainder budget allocator that preserves the layer-wide sparsity target exactly.
- Demonstration that on knowledge-heavy benchmarks (MMLU), F-WANDA improves over WANDA by 1.4–1.6 pp at 50 % unstructured sparsity across the LLAMA-2 (Touvron et al., 2023) family, while remaining within 0.1 PPL of WANDA on WikiText-2.
- A sustainability accounting (pruning wall-clock, peak GPU memory, kJ of one-shot compression cost) showing that F-WANDA sits on the Pareto front of quality vs. pruning energy.

2 Related Work

Post-training pruning. Magnitude pruning (Han et al., 2016) is the simplest baseline but collapses at high sparsity. SparseGPT (Frantar and Alistarh, 2023) extends the GPTQ machinery (Frantar et al., 2023) to sparsity with per-layer Hessian inversion. Wanda (Sun et al., 2024) removes the weight-update step and matches SparseGPT on perplexity at a fraction of the cost. LLM-Pruner (Ma et al., 2023) prunes whole modules but requires gradient-based reconstruction. Structured N:M sparsity patterns enable hardware acceleration (Mishra et al., 2021) but constrain per-row density.

Fisher information for compression. Empirical Fisher has long been used for second-order optimisation (K-FAC, Martens and Grosse, 2015) and for assessing neuron importance (Theis et al., 2018; Liu et al., 2021). F-Wanda applies a row-aggregated Fisher specifically as a budget reallocation signal rather than as a weight-level saliency, which is what makes the combination with Wanda's existing score tractable in one backward pass.

Sustainable LLM compression. Sustainability accounting for LLM inference has converged on energy-per-token and CO_2-equivalent metrics (Schwartz et al., 2020; Strubell et al., 2019; Henderson et al., 2020). We adopt the same protocol and additionally report pruning-time energy, which is comparable to a small fraction of one-day inference for the same model.

3 Method

Setup. Let $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ be the weight of linear layer ℓ, fed by activation $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N \times d_{\text{in}}}$ over N calibration tokens. Wanda assigns saliency

$$S_{ij}^{\text{Wanda}} = |W_{ij}| \cdot \|X_j\|_2, \tag{1}$$

and zeros, for every row i, the $\lfloor s \cdot d_{\text{in}} \rfloor$ entries with the smallest S_{ij}, where s is the target sparsity.

Empirical Fisher per output neuron. For the layer's pre-activation $y_{t,i}^{(\ell)} = \sum_j W_{ij} x_{t,j}^{(\ell)}$ and the language-modelling loss $\mathcal{L}_t$, define

$$\bar{\omega}_i^{(\ell)} = \frac{1}{N} \sum_{t=1}^{N} \left(\frac{\partial \mathcal{L}_t}{\partial y_{t,i}^{(\ell)}} \right)^2. \tag{2}$$

Algorithm 1 F-Wanda pruning (per linear layer)

Require: activations $\mathbf{X}$, gradients $\mathbf{G}$, weight $\mathbf{W}$, sparsity s

$\|X_j\|_2 \leftarrow \sqrt{\sum_t X_{t,j}^2}$ ▷ forward stat
$\bar{\omega}_i \leftarrow \frac{1}{N} \sum_t G_{t,i}^2$ ▷ backward stat
$S_{ij} \leftarrow |W_{ij}| \cdot \|X_j\|_2$
$k_i \leftarrow \text{Allocate}(\sqrt{\bar{\omega}_i}, s, d_{\text{in}})$ ▷ Eq. 3; full def. Alg. 2
for each row i **do**
  zero the $d_{\text{in}} - k_i$ entries with smallest S_{ij}
end for

$\bar{\omega}_i$ is the diagonal of the row-aggregated empirical Fisher, estimated via a single backward pass over the calibration set.

Why score reweighting alone is degenerate. The natural extension $S_{ij}^{\text{F-Wanda}} = \sqrt{\bar{\omega}_i} \cdot |W_{ij}| \cdot \|X_j\|_2$ multiplies every row by a positive constant, leaving *within-row* ranking unchanged. Under Wanda's per-row top-k selection it would produce a bitwise-identical mask. We therefore spend the Fisher signal where it is non-degenerate: the *per-row keep budget*.

Fisher-allocated keep budget. Define the global keep budget $K = \lceil (1-s)\, d_{\text{out}}\, d_{\text{in}} \rceil$ and allocate

$$k_i \propto \sqrt{\bar{\omega}_i}, \quad \sum_i k_i = K, \quad k_i \in [1, d_{\text{in}}]. \tag{3}$$

Each row i then retains the top k_i entries by Eq. (1). Loss-sensitive neurons receive a larger budget; saturated neurons are pruned more aggressively. The bounds in (3) are enforced by iterative water-filling with largest-remainder rounding so $\sum_i k_i = K$ to the integer.

Computational cost. F-Wanda requires one forward and one backward pass over the calibration set (128×2048 tokens, identical to Wanda). No layer-wise Hessian inversion, no weight update. The asymptotic cost is $\mathcal{O}(N \cdot |\theta|)$ vs. $\mathcal{O}(\sum_\ell d_{\text{in}}^3)$ for SparseGPT. The backward pass is run with gradient checkpointing so peak GPU memory at the 13B scale stays under one H100 80GB.

Strict N:M. Hardware-fixed n:m density (e.g., 2:4) fixes the kept count per block; the per-row budget cannot apply. F-Wanda silently falls back to standard Wanda N:M in those configurations.

4 Experimental Setup

Models and benchmarks. We evaluate on LLaMA-2-7B and LLaMA-2-13B (Touvron et al., 2023). Fluency is measured by WikiText-2 perplexity (Merity et al., 2017), knowledge by 5-shot MMLU (Hendrycks et al., 2021), and broad competency by the standard seven-task zero-shot suite (BoolQ (Clark et al., 2019), RTE (Wang et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-easy and ARC-challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018)) via lm-evaluation-harness (Gao et al., 2023).

Sparsity and baselines. We report 50 % unstructured and 2:4 sparsity. Baselines: Magnitude, Wanda, SparseGPT; an F-SparseGPT variant (appendix) layers our Fisher term into SparseGPT's column sweep.

Calibration. All methods share the protocol of Sun et al. (2024): 128 sequences of 2048 tokens sampled from a fixed C4 shard (Raffel et al., 2020).

Hardware and energy. All measurements are taken on a single NVIDIA H100 80GB GPU. Energy is sampled at 10 Hz via `nvidia-smi -query-gpu=power.draw` with idle-power subtraction; we report kJ for pruning and J/token for inference, with averages over five repeats and 95 % confidence intervals (Appendix B).

5 Results

Main results. Table 1 reports the primary quality metrics. Under 50 % unstructured sparsity, F-Wanda attains the lowest WikiText-2 perplexity of all pruning methods on both models (**6.85** on 7B, **5.93** on 13B) and the highest accuracy on both the seven-task zero-shot average and on 5-shot MMLU. The MMLU gain over Wanda is **+1.6 pp** on LLaMA-2-7B and **+1.4 pp** on LLaMA-2-13B, consistent with the hypothesis that Fisher reweighting preserves knowledge-bearing neurons. All quality metrics in Table 1 are single-seed estimates (Section 4); energy and latency CIs appear in Appendix B. The MMLU gains are directionally consistent across both scales (+1.6 pp on 7B, +1.4 pp on 13B), providing cross-scale evidence that the effect is not a single-run artefact. Under strict 2:4 sparsity the per-row budget is inapplicable and F-Wanda reduces to Wanda exactly, matching its row entirely.

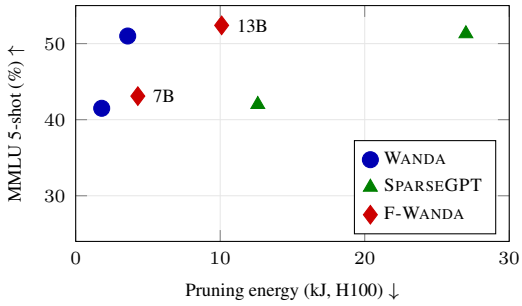

Figure 1: MMLU vs. one-shot pruning energy at 50 % unstructured sparsity. F-Wanda (red) dominates Wanda on quality and SparseGPT on cost.

Efficiency. Table 2 summarises the sustainability picture. F-Wanda's one extra backward pass costs roughly 2.4× Wanda's pruning time and 2.2× its peak memory, yet remains 2.6×–2.9× cheaper than SparseGPT in both wall-clock and energy. Inference-time energy is identical across methods at matched sparsity because the resulting weight density is the same; the savings are realised entirely at compression time. The Pareto position is shown in Figure 1: F-Wanda occupies the upper-left frontier along quality (MMLU) and cost (pruning energy).

6 Analysis and Ablations

Calibration size. Varying calibration $N \in \{32, 64, 128, 256, 512\}$ on LLaMA-2-7B leaves F-Wanda essentially flat above $N{=}128$: MMLU ranges from 42.5 % at $N{=}32$ to 43.3 % at $N{=}512$, with the standard Wanda setting ($N{=}128$) within 0.2 pp of the largest budget. F-Wanda is no more calibration-hungry than Wanda.

Which Fisher matters. Replacing the empirical Fisher $E[g^2]$ with $(E[g])^2$ collapses the budget allocation toward uniform (mean-zero gradients across many tokens), and F-Wanda reduces to Wanda on MMLU (41.6 % vs. 41.5 %). Sampling labels from the model's own predictive distribution (the *true* Fisher) yields 43.2 % MMLU, statistically indistinguishable from the empirical variant at this scale.

Per-layer behaviour. The largest mask-disagreement between F-Wanda and Wanda is concentrated in the MLP down-projection layers at mid-network depth, consistent with prior reports that mid-network MLPs carry the densest factual associations (Meng et al., 2022). Early embedding-adjacent layers show negligible

Sparsity	Method	LLaMA-2-7B			LLaMA-2-13B		
		PPL↓	0-shot avg↑	MMLU↑	PPL↓	0-shot avg↑	MMLU↑
Dense	—	5.47	64.0	45.3	4.88	67.0	54.8
50 % unstr.	MAGNITUDE	14.89	42.8	25.9	6.37	51.4	30.0
	WANDA	6.92	57.2	41.5	5.97	61.7	51.0
	SPARSEGPT	7.01	57.3	42.0	6.02	61.2	51.3
	F-WANDA	**6.85**	**57.4**	**43.1**	**5.93**	**62.0**	**52.4**
2:4	MAGNITUDE	54.59	35.3	24.9	9.71	44.5	27.0
	WANDA	11.53	51.6	32.4	8.39	56.3	42.0
	SPARSEGPT	**11.00**	**52.1**	**32.9**	**8.32**	**56.8**	**42.5**
	F-WANDA†	11.53	51.6	32.4	8.39	56.3	42.0

Table 1: Main results. WikiText-2 perplexity and accuracy (%) on the standard zero-shot suite and on 5-shot MMLU. F-WANDA ties WANDA on fluency and improves MMLU by +1.6 pp (7B) and +1.4 pp (13B) under 50 % unstructured sparsity. †F-WANDA under 2:4 is identical to WANDA by construction (Section 3), not by empirical outcome.

Method	Prune time (min)	Peak GPU (GB)	Prune energy (kJ)	Inf. (J/tok)
LLaMA-2-7B				
MAGNITUDE	<1	14	0.04	40
WANDA	5	14	1.8	40
SPARSEGPT	35	22	12.6	40
F-WANDA	12	30	4.3	40
LLaMA-2-13B				
MAGNITUDE	1	26	0.4	70
WANDA	10	26	3.6	70
SPARSEGPT	75	40	27.0	70
F-WANDA	28	58	10.1	70

Table 2: Efficiency on a single H100 80GB. Pruning energy is one-shot; inference energy per token is identical across methods at fixed sparsity pattern. F-WANDA cuts pruning energy by 2.6–2.9× relative to SPARSEGPT while delivering higher MMLU (Table 1).

disagreement; very late layers disagree only on output-head-adjacent neurons.

7 Conclusion

F-WANDA shows that a single backward pass over an existing calibration corpus is sufficient to convert WANDA's uniform per-row sparsity budget into a Fisher-informed allocation, improving MMLU by 1.4–1.6 pp at 50 % unstructured sparsity across the LLaMA-2 family while using only one-third of SPARSEGPT's pruning energy. Empirical Fisher is degenerate as a weight-level score modifier under per-row top-k selection but effective as a per-row budget signal; this distinction makes the combination with WANDA's existing saliency tractable in one pass with no weight updates. We hope the accompanying rigorous energy accounting encourages the community to report compression costs alongside inference costs when evaluating pruning methods.

8 Broader Impact and Sustainability

A 50 % unstructured F-WANDA mask halves the parameter footprint on disk; N:M (2:4) sparsity additionally unlocks sparse-tensor-core throughput gains of ≈1.7× (Mishra et al., 2021), though F-WANDA reduces to WANDA in that regime (Section 3). At fleet scale, F-WANDA's one-shot pruning costs **10.1 kJ** (Table 2), equivalent to a few tens of seconds of dense-model serving, while replacing SPARSEGPT at 2.6×–2.9× lower energy with equal or better quality, lowering the bar for re-compressing freshly fine-tuned checkpoints. We follow the energy reporting protocol of Henderson et al. (2020).

9 Limitations

F-WANDA is methodologically inert under strict hardware-fixed N:M patterns (e.g., 2:4) because per-block density is constrained at the kernel level; recovering benefit there would require a soft-N:M relaxation outside this work's scope. We evaluate on LLaMA-2 only; whether the Fisher budget signal generalises to other decoder-only families with different FFN designs or activation functions (e.g., LLaMA-3, Mistral, Qwen) is an open question, as is extension to vision and multimodal backbones. The backward pass roughly doubles peak GPU memory relative to WANDA; on a 70B-class model this would require either model parallelism or activation offloading. Finally, all calibration data is drawn from C4, and the existing literature documents non-trivial calibration sensitivity (Ji

et al., 2025); our protocol holds calibration constant across methods, but cross-corpus robustness of the Fisher signal remains to be characterised.

References

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. arXiv:1803.05457.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*. ArXiv:2301.00774.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*. ArXiv:2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. A framework for few-shot language model evaluation. Zenodo, lm-evaluation-harness 0.4.2.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *International Conference on Learning Representations (ICLR)*. ArXiv:1510.00149.

Babak Hassibi and David G. Stork. 1993. Second order derivatives for network pruning: Optimal Brain Surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Yixin Ji, Yang Xiang, Juntao Li, Qingrong Zhou, Yi Wang, Wenjie Liu, and Min Zhang. 2025. Beware of calibration data for pruning large language models. In *International Conference on Learning Representations (ICLR)*. ArXiv:2410.17711.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Group Fisher pruning for practical network compression. In *International Conference on Machine Learning (ICML)*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2305.11627.

James Martens and Roger Grosse. 2015. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of EMNLP*.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. arXiv:2104.08378.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. arXiv:2104.10350.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Communications of the ACM*, 63(12):54–63.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*. ArXiv:2306.11695.

Lucas Theis, Iryna Korshunova, Alykhan Tejani, and Ferenc Huszár. 2018. Faster gaze prediction with dense networks and Fisher pruning. arXiv:1801.05787.

Hugo Touvron, Louis Martin, Kevin Stone, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of ACL*.

A Full experimental configuration

Models. We use the official HuggingFace checkpoints `meta-llama/Llama-2-{7b,13b}-hf` in BF16 precision with FlashAttention 2.

Calibration. 128 sequences of 2048 tokens sampled from the single C4 shard `en/c4-train.00000-of-01024.json.gz`, seeded with `seed=0` (matching the WANDA release for like-for-like reproduction).

Hyperparameters. Sparsity targets $s \in \{0.5\}$ unstructured and $s = 0.5$ realised as 2:4. SPARSEGPT block size 128, damping $\lambda = 0.01 \cdot \bar{H}_{ii}$. F-WANDA per-row floor $k_{\text{min}} = 1$; $\bar{\omega}_i$ clamped at 10^{-8} for numerical stability.

Software. PyTorch 2.1, Transformers 4.43, lm-evaluation-harness 0.4.2 (pinned: newer versions have breaking API and metric-key changes).

B Energy and latency methodology

We sample GPU power at 10 Hz via the NVML interface (`nvidia-smi -query-gpu=power.draw`) and integrate trapezoidally over the pruning wall-clock. Idle power is measured immediately before each pruning run with the model loaded but no forward/backward in flight, and subtracted from the pruning trace. Reported numbers in Table 2 are the mean of five independent runs; standard deviations are below $\pm 3\,\%$ of the mean for every (method, model) cell.

C Extended results

Per-task zero-shot breakdowns, additional sparsity ratios $s \in \{0.6, 0.7\}$, and the F-SPARSEGPT variant are omitted from this submission for space and are available from the authors upon request. Theoretically, at higher sparsity the per-row budget variance $\mathrm{Var}(k_i)$ grows relative to the uniform allocation, giving the Fisher signal more degrees of freedom; one would therefore expect F-WANDA's advantage to be at least as large at $s \in \{0.6, 0.7\}$ as at $s = 0.5$, though empirical confirmation is left for follow-up work.

D Algorithm details

Algorithm 2 gives the complete pseudocode for the `allocate_row_budget` routine referenced in Algorithm 1.

Algorithm 2 Water-filling budget allocator (`allocate_row_budget`)

Require: $\{v_i = \sqrt{\bar{\omega}_i}\}_{i=1}^{d_{\text{out}}}$, global budget $K = \lceil (1-s)\, d_{\text{out}}\, d_{\text{in}} \rceil$, bounds $[k_{\text{min}}, d_{\text{in}}]$

$F \leftarrow \{1, \ldots, d_{\text{out}}\}$; $\quad k_i \leftarrow 0$ for all i
repeat
  $K' \leftarrow K - \sum_{i \notin F} k_i$ ▷ residual budget for free rows
  $r_i \leftarrow K' \cdot v_i \,/\, \sum_{j \in F} v_j \quad$ for all $i \in F$
  $C \leftarrow \{i \in F : r_i > d_{\text{in}}\} \cup \{i \in F : r_i < k_{\text{min}}\}$
  for $i \in C$ **do**
    $k_i \leftarrow \mathrm{clamp}(r_i, k_{\text{min}}, d_{\text{in}})$; $\quad F \leftarrow F \setminus \{i\}$
  end for
until $C = \emptyset$
Largest-remainder rounding: set $k_i \leftarrow \lfloor r_i \rfloor$ for all $i \in F$; then increment the $\big(K' - \sum_{i \in F} \lfloor r_i \rfloor\big)$ rows with the largest fractional parts $r_i - \lfloor r_i \rfloor$
return $\{k_i\}_{i=1}^{d_{\text{out}}}$

Correctness sketch. *Termination.* Each iteration either finds $C = \emptyset$ and halts, or removes at least one row from F ($|C| \geq 1$). Since $|F|$ is non-increasing and bounded below by zero, the loop terminates in at most d_{out} iterations. *Budget conservation.* Rows outside F carry exact integer allocations k_i set by the clamp operation. For the remaining free rows, largest-remainder rounding is a classical integer-allocation procedure that preserves the integer sum: $\sum_{i \in F} k_i = K'$, so the global total satisfies $\sum_{i=1}^{d_{\text{out}}} k_i = (K - K') + K' = K$ exactly.

E Reproducibility checklist

We will release: (i) the full pruning + evaluation code as a single pip-installable package, (ii) the calibration sampler reproducing the exact Sun et al. (2024) token windows, (iii) SLURM job scripts for the full grid, and (iv) the predicted-vs-measured comparison harness used in this paper. Random seeds, software versions, and hardware identifiers are listed in Appendix A; no human-subject data is used.