AlphaDecay: Module-wise Weight Decay for Heavy-Tailed Balancing in LLMs

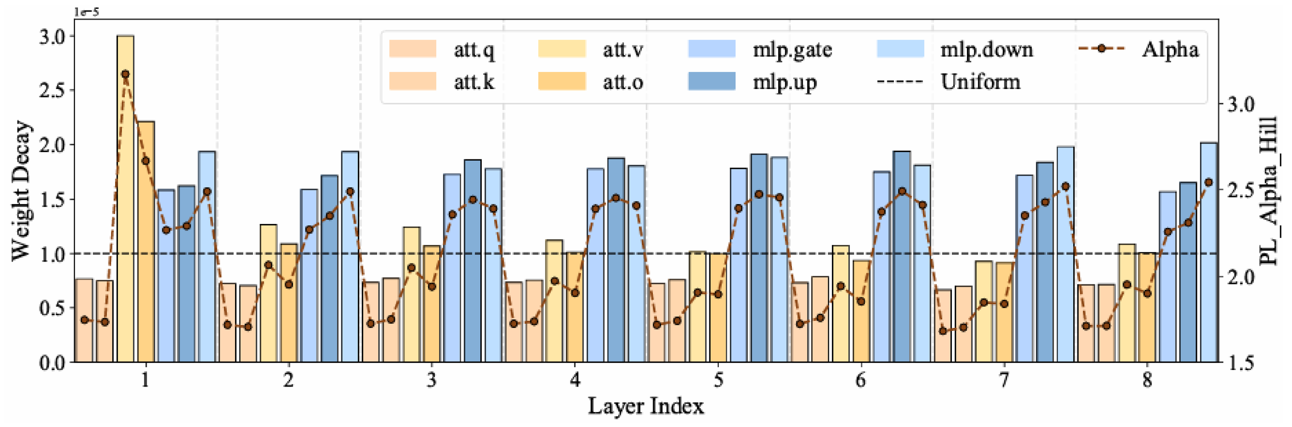
Di He[1,2,3], Songjun Tu[2,3], Ajay Jaiswal[4], Li Shen[5] , Ganzhao Yuan[6], Shiwei Liu[7], Lu Yin[8*]

[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences , [2]Peng Cheng Laboratory, [3]University of Chinese Academy of Sciences [4]University of Texas at Austin
[5]Shenzhen Campus of Sun Yat-sen University, [6]Shenzhen University of Advanced Technology [7]University of Oxford [8]University of Surrey

Paper & Code

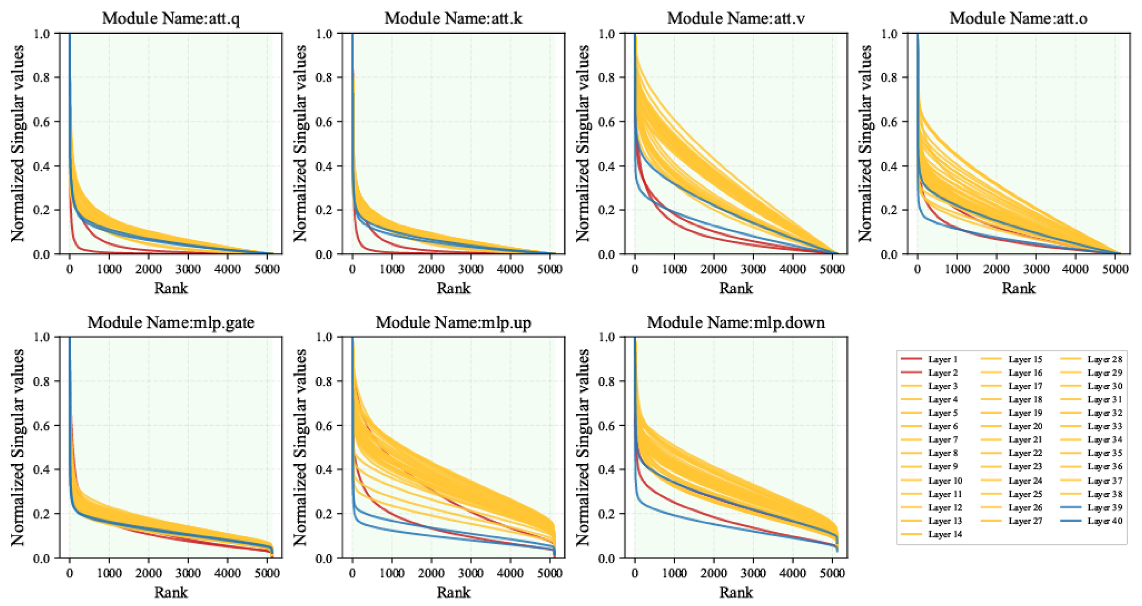
Research Question

All popular optimizers use uniform weight decay—does a better configuration exist for LLMs?



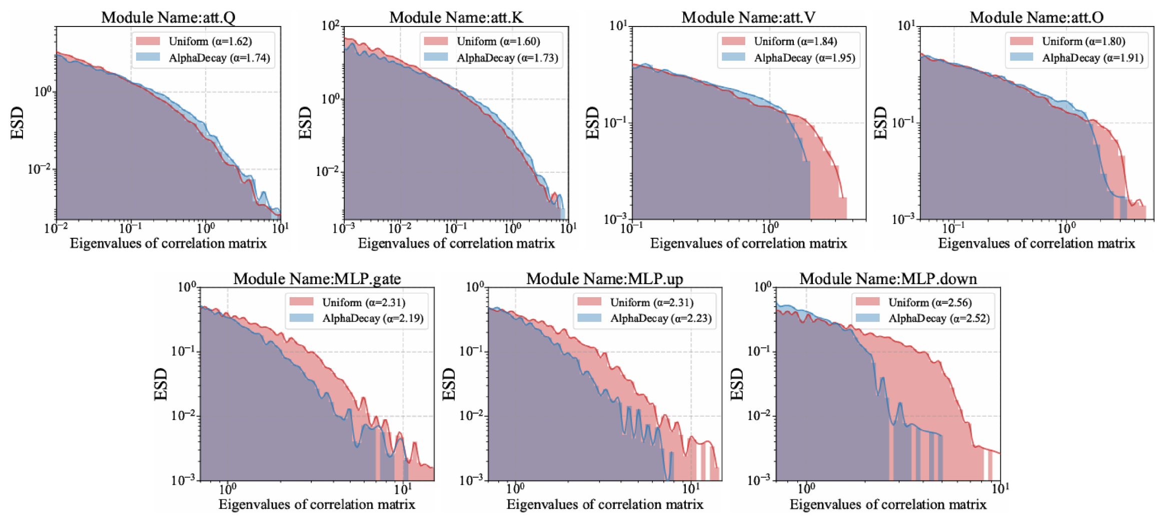
Rationale

Modules display distinct spectral characteristics — attention layers are heavier-tailed, while MLP layers are lighter-tailed.



Balance works?

AlphaDecay benefits training, yielding lower perplexity (22.55 < 23.14).



Method: AlphaDecay

More heavier-tailed, Lower Weight Decay!

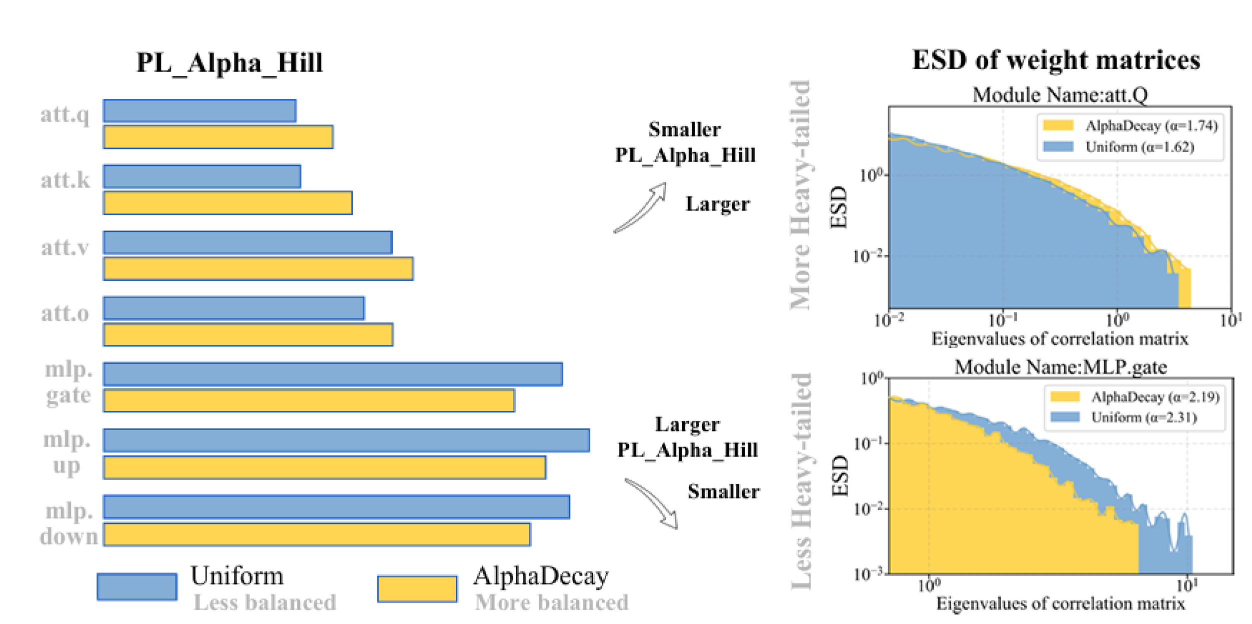
Power law distribution ,

$$p(\lambda) \propto \lambda^{-\alpha}, \lambda_{\min} < \lambda < \lambda_{\max}$$

Spectral propertie:

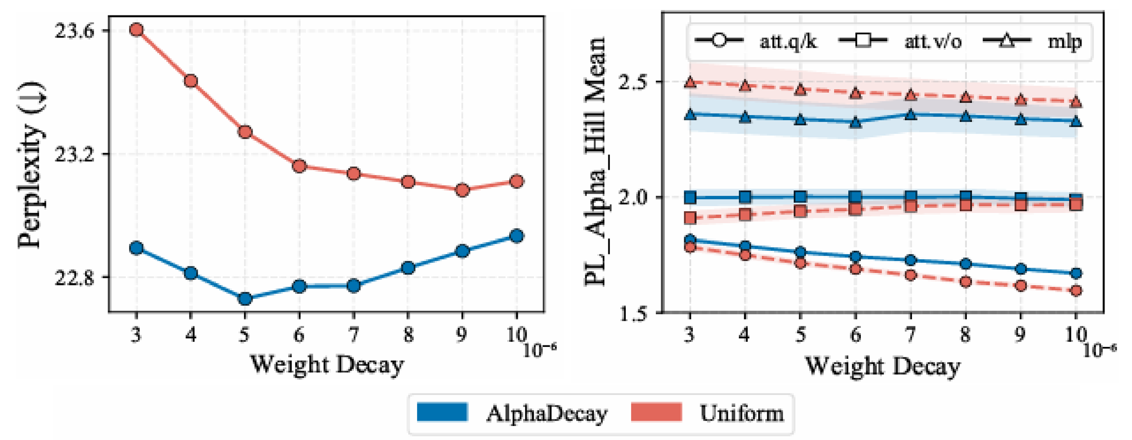
$$\texttt{PL\_Alpha\_Hill} = 1 + \frac{k}{\sum_{i=1}^{k} ln \frac{\lambda_{n-i+1}}{\lambda_{n-k}}}$$

λ: eigenvalues of the correlation matrix
k: the lower cutoff for PL fitting



Spectrum Balanced

AlphaDecay balances spectra and improves performance.



Results

Table 2: **(Main result).** Comparison with various weight decay scheduling strategies on pre-training various sizes of LLaMa models on C4 dataset. Validation perplexity (↓) is reported. All baselines are carefully tuned. 'WD=0' indicates that weight decay is disabled during model training.

	LLaMa-60M			**LLaMa-135M**			**LLaMa-350M**			**LLaMa-1B**		
Weight Decay	1e-5	5e-6	1e-6	1e-5	5e-6	1e-6	1e-5	5e-6	1e-6	1e-5	5e-6	1e-6
WD=0		33.23			24.60			18.62			16.11	
Uniform	32.39	32.56	33.03	22.99	23.14	24.14	17.12	16.74	17.50	15.36	14.66	15.03
AWD[12]	33.78	33.74	33.74	24.25	24.45	24.53	18.32	18.55	18.79	16.03	16.22	16.38
Adadecay[30]	32.24	32.52	33.03	23.20	23.08	23.96	18.21	17.42	17.91	17.23	18.14	15.35
AlphaDecay	**31.56**	**31.58**	**32.61**	**22.76**	**22.55**	**23.49**	**17.00**	**16.66**	**16.88**	**15.13**	**14.55**	**14.63**

AlphaDecay is plug-and-play, task-agnostic, optimizer-agnostic, and nearly cost-free.

- **Zero-shot performance**

Table 3: **(Zero-shot results of commonsense-reasoning tasks).** Zero-shot evaluation results (↑) on seven commonsense reasoning benchmarks using the LLaMa-1B model pretrained with different methods.

Method	ARC-c	ARC-e	PIQA	Hellaswag	OBQA	Winogrande	BOOLQ	Avg.
Uniform	20.22	46.72	67.68	32.94	18.8	49.41	54.74	41.50
AdaDecay	19.20	46.72	66.97	32.96	18.0	**51.54**	56.36	41.68
AWD	19.18	46.34	66.65	31.37	18.0	51.07	57.25	41.41
AlphaDecay	**20.90**	**48.86**	**68.44**	**34.16**	**19.80**	50.59	**60.70**	43.35

- **Pre-training with AdamW**

Table 12: **(AdamW.)** Comparison of various weight decay scheduling strategies using AdamW optimizer for pre-training LLaMa-60M and LLaMa-130M models under different weight decay values. Validation perplexity (↓) on the C4 dataset is reported. All baselines are carefully tuned. 'WD=0' indicates that weight decay is disabled during model training.

	LLaMa-60M			LLaMa-135M		
Weight Decay	0.1	0.05	0.01	0.1	0.05	0.01
WD=0		32.73			24.39	
Uniform	31.95	32.31	32.66	23.32	23.81	24.28
AWD	32.58	32.67	32.67	24.30	24.35	24.41
Adadecay	31.88	32.25	32.58	23.18	23.62	24.21
AlphaDecay	31.20	31.65	32.45	22.66	23.04	23.98

- **Finetuning tasks**

Table 4: **(Finetuning tasks).** Finetuning results (↑) on eight benchmarks from the GLUE dataset using `roberta-base` with different methods.

Method	cola	mnli	mrpc	qnli	qqp	rte	sst2	stsb	Avg.
Uniform	59.73	86.78	87.01	92.59	89.97	70.11	93.69	90.78	83.83
AdaDecay	60.45	87.23	88.19	92.62	89.95	73.36	93.73	90.9	84.55
AWD	60.72	**87.44**	89.53	92.58	90.08	72.27	93.72	90.9	84.66
AlphaDecay	**62.82**	87.11	**89.61**	**92.73**	**90.12**	**73.86**	**93.77**	**90.91**	**85.12**

- **AlphaDecay cost**

Table 11: Parameter settings of the experiment reported in Section 4.4 Figure 8. The computation times reflect the NVIDIA A100 hours utilized for completing model training.

Model Size	Weight Decay	Uniform	GAP=500	GAP=250	GAP=100	GAP=50	GAP=1	Scaling Ratio (s_1, s_2)
LLaMa -60M	1e-5	32.386	31.614	31.628	**31.555**	31.618	31.594	(0.67,3)
	5e-6	32.562	**31.628**	31.633	31.673	31.717	31.712	(0.67,5)
	1e-6	33.029	32.703	32.718	32.754	**32.663**	32.769	(0.67,5)
	Computation Time	1.4h	1.4h	1.4h	1.5h	1.6h	9.3h	
LLaMa -135M	1e-5	22.994	22.763	**22.756**	22.779	22.758	22.809	(0.67,3)
	5e-6	23.138	22.551	**22.537**	22.569	22.539	22.581	(0.67,5)
	1e-6	24.142	23.488	23.477	23.479	**23.468**	23.488	(0.67,5)
	Computation Time	5.6h	5.7h	5.9h	6.3h	7.1h	74.5h	

- **Across architectures and datasets**

Table 5: **(Across architectures and datasets).** Results on GPT-nano/C4 (Perplexity) and ViT-tiny/ImageNet-1K (Top-1) with different methods.

Backbone / Dataset	Metric	Uniform	AWD	AdaDecay	AlphaDecay
GPT-nano / C4	PPL(↓)	27.56	27.64	27.68	**27.37**
ViT-tiny / ImageNet-1K	Top-1(↑)	66.41%	64.98%	66.26%	**67.73%**