A THEORETICAL PROOFS

A.1 PROOF OF THEOREM 1

Proof. Fix w_{ij} and set $f(\alpha) \equiv f_{ij}(\alpha) = g_{ij}(\alpha) = \partial \mathcal{L}(\alpha \Delta \mathbf{W})/\partial w_{ij}$. By Eq. (4), $s_e(w_{ij}) = |w_{ij}| \left| \int_0^1 f(\alpha)\, d\alpha \right|$. Define the composite trapezoidal approximation and its sampled variant:

$$\mathcal{T}_N = \frac{1}{2N}\left[f(0) + 2\sum_{k=1}^{N-1} f\big(\tfrac{k}{N}\big) + f(1) \right], \qquad \widetilde{\mathcal{T}}_M = \frac{1}{2N}\left[f(0) + 2(N-1)\,\overline{f}_M + f(1) \right], \quad (13)$$

where $\overline{f}_M = \frac{1}{M}\sum_{p=1}^{M} f(\alpha_p)$ with α_p i.i.d. drawn from the discrete uniform distribution on $\{1/N, \ldots, (N-1)/N\}$.

Since $s_{agg}(w_{ij}) = |w_{ij}|\,|\widetilde{\mathcal{T}}_M|$ and $||x| - |y|| \le |x - y|$, the triangle inequality yields

$$\left| s_e(w_{ij}) - s_{agg}(w_{ij}) \right| \;\le\; |w_{ij}| \left| \int_0^1 f - \widetilde{\mathcal{T}}_M \right| \;\le\; |w_{ij}| \Big(\left| \int_0^1 f - \mathcal{T}_N \right| + |\mathcal{T}_N - \widetilde{\mathcal{T}}_M| \Big). \quad (14)$$

Step 1: discretization error. By assumption, f is twice continuously differentiable on $[0,1]$ and $\sup_{\alpha\in[0,1]} |f''(\alpha)| \le C_2$. The standard error bound for the composite trapezoidal rule on $[0,1]$ (see, e.g., classical numerical analysis texts) yields

$$\left| \int_0^1 f(\alpha)\, d\alpha - \mathcal{T}_N \right| \;\le\; \tfrac{C_2}{12\,N^2}. \quad (15)$$

Step 2: sampling error. Let $\mu = \frac{1}{N-1}\sum_{k=1}^{N-1} f(\frac{k}{N})$ denote the average of f over the $(N-1)$ interior nodes. A simple algebraic manipulation gives

$$|\mathcal{T}_N - \widetilde{\mathcal{T}}_M| = \frac{1}{N}\Big| \sum_{k=1}^{N-1} f\big(\tfrac{k}{N}\big) - (N-1)\,\overline{f}_M \Big| = \frac{N-1}{N}\,|\mu - \overline{f}_M| \;\le\; |\mu - \overline{f}_M|. \quad (16)$$

By assumption, $f(\alpha)$ is uniformly bounded on the discretization nodes, which is discussed in detail in Appendix B.1: there exists $B < \infty$ such that $|f(\alpha)| \le B$ for all $\alpha \in \{1/N, \ldots, (N-1)/N\}$. Therefore, each sample $f(\alpha_p)$ lies in $[-B, B]$, and Hoeffding's inequality for bounded random variables implies that, for any $\delta \in (0,1)$,

$$\Pr\big(|\mu - \overline{f}_M| \ge t\big) \;\le\; 2\exp\!\left(-\frac{2Mt^2}{(2B)^2}\right) = 2\exp\!\left(-\frac{Mt^2}{2B^2}\right). \quad (17)$$

Setting the right-hand side equal to δ and solving for t yields that, with probability at least $1-\delta$,

$$|\mu - \overline{f}_M| \;\le\; B\sqrt{\frac{2\log(2/\delta)}{M}} \;\le\; c\,B\sqrt{\frac{\log(1/\delta)}{M}} \quad (18)$$

for an absolute constant $c > 0$. Combining with the previous display gives

$$|\mathcal{T}_N - \widetilde{\mathcal{T}}_M| \;\le\; |\mu - \overline{f}_M| \;\le\; c\,B\sqrt{\frac{\log(1/\delta)}{M}} \quad (19)$$

with probability at least $1-\delta$.

Step 3: combining the bounds. Plugging Eq. (15) and Eq. (19) into the decomposition in Eq. (14) yields that, with probability at least $1-\delta$,

$$\left| s_e(w_{ij}) - s_{agg}(w_{ij}) \right| \;\le\; |w_{ij}| \Big(\frac{C_2}{12\,N^2} + c\,B\sqrt{\frac{\log(1/\delta)}{M}} \Big), \quad (20)$$

which is exactly the claimed bound in Eq. (9). □

A.2 HIGH-PROBABILITY STABILITY OF SNR_t

The resulting SNR-based score favors parameters with consistent, high-impact contributions and suppresses those with volatile or transient behavior. While the above formulation provides an intuitive interpretation of SNR, it remains essential to ensure its statistical stability with high probability, which is formally addressed in Theorem 2.

Theorem 2. *Let $y_t = s_{agg}(w_{ij})$ be the per-epoch raw importance defined in Eq. (7). Since ϵ in Eq. (12) is a very small constant, it can be ignored. Therefore, we have:*

$$\mathsf{SNR}_t = \frac{\bar{s}_e^{(t)}}{\bar{U}^{(t)} + \epsilon} \approx \frac{\bar{s}_e^{(t)}}{\bar{U}^{(t)}}, \tag{21}$$

Assume that (y_t) is an i.i.d. sequence of sub-Gaussian random variables with mean μ and variance σ^2, and let $d = \mathbb{E}\big[|y_t - \mu|\big] > 0$. For $\beta_1, \beta_2 \in (0, 1)$, define the effective EMA window lengths

$$n_{\text{eff}}(\beta_1) = \frac{1+\beta_1}{1-\beta_1}, \quad n_{\text{eff}}(\beta_2) = \frac{1+\beta_2}{1-\beta_2}, \quad n_{\text{eff}} = \min\{n_{\text{eff}}(\beta_1), n_{\text{eff}}(\beta_2)\}. \tag{22}$$

Then there exist universal constants $c_1, c_2, c_0 > 0$ such that, for any $\delta \in (0, 1)$ and all

$$t \;\geq\; t_{\text{burn}} \;=\; \left\lceil \frac{c_1}{1 - \min\{\beta_1, \beta_2\}} \log \frac{c_2}{\delta} \right\rceil, \tag{23}$$

the following holds with probability at least $1 - \delta$:

$$\big|\mathsf{SNR}_t - \mu/d\big| \;\leq\; C \sqrt{\frac{\log(2/\delta)}{n_{\text{eff}}}}, \qquad C \;=\; \frac{2\sqrt{2}\,\sigma}{d} + 2c_0\,\frac{\mu}{d^2}\,(\sigma + d). \tag{24}$$

Proof. We analyze the EMA under the stylized assumption stated in Theorem 2: (y_t) is an i.i.d. sub-Gaussian sequence with mean μ, variance proxy σ^2, and $d = \mathbb{E}|y_t - \mu| > 0$.

Recall that Eq. (10) and Eq. (11) define the EMAs

$$\bar{s}_e^{(t)} = \beta_1 \bar{s}_{t-1} + (1-\beta_1) y_t, \qquad \bar{U}^{(t)} = \beta_2 \bar{U}_{t-1} + (1-\beta_2)\,\big|y_t - \bar{s}_e^{(t)}\big|. \tag{25}$$

Unrolling the recursions (for t large enough so that transients are negligible) shows that

$$\bar{s}_e^{(t)} = \sum_{k \geq 0} w_k^{(1)}\, y_{t-k}, \quad w_k^{(1)} = (1-\beta_1)\beta_1^k, \qquad \bar{U}^{(t)} = (1-\beta_2) \sum_{k \geq 0} \beta_2^k \,\big|y_{t-k} - \bar{s}_{t-k}\big|. \tag{26}$$

Note that $(w_k^{(1)})_{k \geq 0}$ is a geometric weight sequence with $\sum_k w_k^{(1)} = 1$ and

$$\|w^{(1)}\|_2^2 = \sum_{k \geq 0} (1-\beta_1)^2 \beta_1^{2k} = \frac{1-\beta_1}{1+\beta_1} = \frac{1}{n_{\text{eff}}(\beta_1)}. \tag{27}$$

Below we write $n_{\text{eff}} = \min\{n_{\text{eff}}(\beta_1), n_{\text{eff}}(\beta_2)\}$.

Step 1: concentration of $\bar{s}_e^{(t)}$. Since (y_t) are i.i.d. sub-Gaussian with mean μ and variance proxy σ^2, any fixed weighted sum $\sum_k w_k^{(1)} y_{t-k}$ is also sub-Gaussian with mean μ and variance proxy $\sigma^2 \|w^{(1)}\|_2^2 = \sigma^2 / n_{\text{eff}}(\beta_1)$. Standard sub-Gaussian tail bounds then yield

$$\Pr\Big(|\bar{s}_e^{(t)} - \mu| \geq \varepsilon\Big) \;\leq\; 2\exp\!\left(-\frac{c\, n_{\text{eff}}(\beta_1)\, \varepsilon^2}{\sigma^2}\right) \tag{28}$$

for an absolute constant $c > 0$. Setting the right-hand side to $\delta/2$ and solving for ε gives

$$|\bar{s}_e^{(t)} - \mu| \;\leq\; \sigma \sqrt{\frac{2\log(4/\delta)}{n_{\text{eff}}(\beta_1)}} \;\leq\; \sqrt{2}\,\sigma \sqrt{\frac{\log(4/\delta)}{n_{\text{eff}}}} \tag{29}$$

with probability at least $1 - \delta/2$.

Step 2: concentration of $\bar{U}^{(t)}$. We decompose $\bar{U}^{(t)}$ around $d = \mathbb{E}|y_t - \mu|$ as

$$|\bar{U}^{(t)} - d| \;\leq\; (1-\beta_2)\Big|\sum_{k \geq 0} \beta_2^k \big(|y_{t-k} - \mu| - d\big)\Big| \;+\; (1-\beta_2) \sum_{k \geq 0} \beta_2^k \big||y_{t-k} - \bar{s}_{t-k}| - |y_{t-k} - \mu|\big|. \tag{30}$$

Define $X_t = |y_t - \mu| - d$, which is a centered, sub-exponential random variable whose tail parameters depend only on (σ, d) (because y_t is sub-Gaussian). Let $w_k^{(2)} = (1-\beta_2)\beta_2^k$ denote the EMA weights for $\bar{U}^{(t)}$. Then $\sum_{k\geq 0} w_k^{(2)} = 1$ and

$$\|w^{(2)}\|_2^2 = \sum_{k\geq 0}(1-\beta_2)^2\beta_2^{2k} = \frac{1-\beta_2}{1+\beta_2} = \frac{1}{n_{\text{eff}}(\beta_2)}.$$

Applying a Bernstein-type concentration for weighted sums of i.i.d. sub-exponential variables (see, e.g., standard results on Orlicz norms) yields the existence of an absolute constant $c_0 > 0$ such that, for any $\delta \in (0,1)$,

$$\Pr\left(\left|(1-\beta_2)\sum_{k\geq 0}\beta_2^k X_{t-k}\right| \geq c_0(\sigma+d)\sqrt{\frac{\log(4/\delta)}{n_{\text{eff}}(\beta_2)}}\right) \;\leq\; \frac{\delta}{2}. \tag{31}$$

For the second term in Eq. (30), note that $\big||a-c| - |a-b|\big| \leq |b-c|$ for any $a, b, c \in \mathbb{R}$, so

$$\big||y_{t-k} - \bar{s}_{t-k}| - |y_{t-k} - \mu|\big| \;\leq\; |\bar{s}_{t-k} - \mu|.$$

Thus

$$(1-\beta_2)\sum_{k\geq 0}\beta_2^k\big||y_{t-k} - \bar{s}_{t-k}| - |y_{t-k} - \mu|\big| \;\leq\; (1-\beta_2)\sum_{k\geq 0}\beta_2^k\,|\bar{s}_{t-k} - \mu|. \tag{32}$$

We now bound the right-hand side by splitting the sum into a recent window and its tail. Let

$$L = \left\lceil \frac{c_1}{1-\beta_2}\,\log\frac{c_2}{\delta}\right\rceil \tag{33}$$

for absolute constants $c_1, c_2 > 0$ chosen large enough. For $t \geq L$, we have

$$(1-\beta_2)\sum_{k\geq 0}\beta_2^k\,|\bar{s}_{t-k} - \mu| \;\leq\; (1-\beta_2)\sum_{k=0}^{L}\beta_2^k\,|\bar{s}_{t-k} - \mu| \;+\; (1-\beta_2)\sum_{k>L}\beta_2^k\,|\bar{s}_{t-k} - \mu|. \tag{34}$$

For the tail sum, $(1-\beta_2)\sum_{k>L}\beta_2^k = \beta_2^{L+1}$ and, by choosing c_1, c_2 appropriately, we can ensure $\beta_2^{L+1} \leq \delta/(8c_2)$. For the finite window $\{t, t-1, \ldots, t-L\}$, we apply Eq. (29) and a union bound over these $(L+1)$ indices to obtain, with probability at least $1 - \delta/2$,

$$|\bar{s}_{t-k} - \mu| \;\leq\; \sqrt{2}\,\sigma\sqrt{\frac{\log(4L/\delta)}{n_{\text{eff}}(\beta_1)}} \quad \text{for all } 0 \leq k \leq L. \tag{35}$$

Combining these bounds and using $n_{\text{eff}} \leq n_{\text{eff}}(\beta_1)$ yields

$$(1-\beta_2)\sum_{k\geq 0}\beta_2^k\,|\bar{s}_{t-k} - \mu| \;\leq\; \tilde{c}\,\sigma\sqrt{\frac{\log(2/\delta)}{n_{\text{eff}}}} \tag{36}$$

with probability at least $1 - \delta/2$, for an absolute constant $\tilde{c} > 0$.

Putting Eq. (31) and Eq. (36) back into Eq. (30) and recalling that $n_{\text{eff}} \leq n_{\text{eff}}(\beta_2)$, we obtain that, for $t \geq t_{\text{burn}}$ and with probability at least $1 - \delta$,

$$|\bar{U}^{(t)} - d| \;\leq\; C_2'(\sigma+d)\sqrt{\frac{\log(2/\delta)}{n_{\text{eff}}}} \tag{37}$$

for an absolute constant $C_2' > 0$. By increasing c_1 if necessary, we may ensure that the right-hand side in Eq. (37) is at most $d/2$, so that $\bar{U}^{(t)} \geq d/2$ holds on the same high-probability event.

Step 3: bounding the ratio SNR_t. On the event $\{\bar{U}^{(t)} \geq d/2\}$ we can control the ratio $\mathsf{SNR}_t = \bar{s}_e^{(t)}/\bar{U}^{(t)}$ via the deterministic inequality

$$\left| \frac{\bar{s}_e^{(t)}}{\bar{U}^{(t)}} - \frac{\mu}{d} \right| \;\leq\; \frac{2}{d}\,|\bar{s}_e^{(t)} - \mu| \;+\; \frac{2\mu}{d^2}\,|\bar{U}^{(t)} - d|. \tag{38}$$

Combining Eq. (29) and Eq. (37) with Eq. (38), and noting that $n_{\text{eff}} \leq n_{\text{eff}}(\beta_1)$, gives

$$\left|\mathsf{SNR}_t - \mu/d\right| \;\leq\; \left(\frac{2\sqrt{2}\,\sigma}{d} + 2c_0\,\frac{\mu}{d^2}(\sigma + d) \right) \sqrt{\frac{\log(2/\delta)}{n_{\text{eff}}}} \tag{39}$$

with probability at least $1 - \delta$, for a suitable absolute constant $c_0 > 0$. This is exactly the claimed bound in Theorem 2 after setting $C = \frac{2\sqrt{2}\,\sigma}{d} + 2c_0\,\frac{\mu}{d^2}(\sigma + d)$ and $t_{\text{burn}} = \left\lceil \frac{c_1}{1 - \min\{\beta_1, \beta_2\}} \log \frac{c_2}{\delta} \right\rceil$. □

B The Discussion of the Assumptions in Theorem

B.1 The Analysis of the Assumption in Theorem 1

In this section, we focus on how the assumption in Theorem 1, that g_{ij} is twice continuously differentiable on the interval $[0, 1]$ with a bounded second derivative, leads to the conclusion that $g_{ij}(\alpha)$ is bounded. First, consider the following form of $g_{ij}(\alpha)$:

$$g_{ij}(\alpha) = \frac{\partial \mathcal{L}(\alpha \Delta \mathbf{W})}{\partial w_{ij}}, \quad \alpha \in [0, 1], \tag{40}$$

The analysis of Theorem 1 relies solely on the assumption that g_{ij} is twice differentiable on the interval $[0, 1]$ and that its second derivative is bounded, which allows the application of the composite trapezoidal rule, leading to a discretization error of $O(N^{-2})$. Specifically, numerical analysis typically assumes the existence of a constant $C_2 < \infty$ such that:

$$\sup_{\alpha \in [0,1]} \left| g_{ij}''(\alpha) \right| \leq C_2. \tag{41}$$

Under this assumption, we can derive the following error bound:

$$\left| \int_0^1 g_{ij}(\alpha)\, d\alpha - \mathcal{T}_N \right| \leq \frac{C_2}{12N^2}, \tag{42}$$

This equation provides the theoretical basis for the $O(N^{-2})$ discretization error term in Theorem 1. This requirement is essentially a standard smoothness assumption in trapezoidal integration and does not involve any specific distributional assumptions. Furthermore, the condition of bounded second derivatives directly implies that g_{ij} itself is bounded. By the fundamental theorem of calculus:

$$g_{ij}'(\alpha) = g_{ij}'(0) + \int_0^\alpha g_{ij}''(t)\, dt,\; g_{ij}(\alpha) = g_{ij}(0) + \int_0^\alpha g_{ij}'(t)\, dt, \tag{43}$$

We can obtain the bound for all $\alpha \in [0, 1]$:

$$|g_{ij}'(\alpha)| \leq |g_{ij}'(0)| + \int_0^1 |g_{ij}''(t)|\, dt \leq |g_{ij}'(0)| + C_2, \tag{44}$$

Thus,

$$|g_{ij}(\alpha)| \leq |g_{ij}(0)| + \int_0^1 |g_{ij}'(t)|\, dt \leq |g_{ij}(0)| + |g_{ij}'(0)| + C_2 \triangleq B. \tag{45}$$

This implies that $g_{ij}(\alpha)$ is bounded on $[0, 1]$. When we sample α from the finite set $\{1/N, \ldots, (N - 1)/N\}$, the resulting random variable $g_{ij}(\alpha)$ is bounded by constant B.

B.2 THE ANALYSIS OF THE I.I.D. ASSUMPTION IN THEOREM 2

Theorem 2 assumes that the per-epoch raw scores $y_t = s_{agg}(w_{ij})$ form an i.i.d. sub-Gaussian sequence with a common mean μ and variance σ^2. However, strictly speaking, y_t depends on the current model parameters $\mathbf{W}^{(t)}$, which are updated across epochs, so exact i.i.d. is an idealization.

Our goal is to model the regime in which the training dynamics have *stabilized*: after an initial transient phase (discarded via the burn-in time t_{burn}), the statistics of the gradient noise around the current solution change only slowly. Furthermore, within the effective EMA window $n_{\text{eff}}(\beta_1, \beta_2)$, the gradient sequence can be approximated as having nearly stationary mean and variance. In this regime, standard extensions of EMA concentration results to weakly dependent or mixing sequences apply. We chose the i.i.d. setting for clarity of presentation and to keep the notation simple. It is important to note that Theorem 2 is derived under this stylized, locally stationary noise assumption, and is meant to provide intuition about how the EMA window size and variance control the stability of SNR_t, rather than to capture every aspect of LLM training dynamics exactly.

To support this approximation empirically, we provide a small diagnostic in Appendix G: for a representative layer on BoolQ, we plot the time series of y_t and its running mean/variance across epochs. We observe that, after the early epochs, both the mean and variance of y_t quickly settle into a narrow band, and the lag-1 autocorrelation becomes small. Correspondingly, the SNR_t curves are nearly flat after burn-in. These observations suggest that, in the regime where EMA-based importance is actually used for rank pruning, the i.i.d./local stationarity approximation is reasonably accurate.

Finally, we emphasize that these assumptions are used only in our theoretical analysis; the algorithm itself does not rely on them. Even when the exact assumptions are relaxed, the qualitative conclusions remain the same: (i) our IG estimator trades off discretization error $O(N^{-2})$ and sampling error $O(M^{-1/2})$, and (ii) EMA-based SNR_t scores become more stable as the effective sample size increases and the process enters a locally stationary regime.

C HYPERPARAMETER SETTINGS

During the training process, we tune the learning rate from $\{5 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-4}\}$ and pick the best learning rate for every method. For the MNLI, QNLI, and QQP, we set the batch size to 128. For RTE, MRPC, CoLA, and STS-B, the batch size is set to 32. For SST-2, we use a batch size of 64. For all other tasks, the batch size is set to 16. All baseline methods follow the same settings as IGU-LoRA, as detailed in Table 6. In IGU-LoRA, several key hyperparameters $\epsilon, M, N, \beta_1, \beta_2$ are set to 1×10^{-6}, 16, 20, 0.85, and 0.85, respectively, as detailed in Table 7. They remain constant throughout the experiment, and their sensitivity is discussed in the main text.

Table 6: Hyperparameter setup of IGU-LoRA for training on different datasets.

Dataset	learning rate	batch size	Max. Sequence Length	# epochs	γ	t_i	Δ_T	t_f
MNLI	5×10^{-4}	128	512	25	0.1	500	20	10000
RTE	1×10^{-3}	32	512	25	0.1	300	5	2500
QNLI	5×10^{-4}	128	512	25	0.1	400	20	10000
MRPC	1×10^{-3}	32	512	25	0.1	300	5	2500
QQP	5×10^{-4}	128	512	25	0.1	500	20	10000
SST-2	1×10^{-3}	64	512	25	0.1	400	20	5000
CoLA	1×10^{-3}	32	512	25	0.1	300	5	2500
STS-B	2×10^{-3}	32	512	25	0.1	300	5	2500
BoolQ	5×10^{-4}	16	512	25	0.1	500	20	10000
ARC-e	5×10^{-4}	16	512	25	0.1	500	20	10000
ARC-c	5×10^{-4}	16	512	25	0.1	500	20	10000
COPA	1×10^{-3}	16	512	25	0.1	500	20	10000
AQuA	1×10^{-4}	16	512	25	0.1	500	20	10000
MMLU	1×10^{-4}	128	512	15	0.1	500	20	10000
VQA	2×10^{-4}	32	512	25	0.1	300	20	10000
GAQ	5×10^{-4}	32	512	25	0.1	300	20	10000
MVLR^2	5×10^{-4}	32	512	25	0.1	300	20	10000
COCO	2×10^{-4}	32	512	25	0.1	300	20	10000

Table 7: Setting of the 5 hyperparameters (ϵ, M, N, β_1, β_2) in IGU-LoRA.

Hyperparameter	ϵ	M	N	β_1	β_2
Value	1×10^{-6}	16	20	0.85	0.85

D Ablation Study on High-Impact Parameters

To further validate the effectiveness of IGU-LoRA in identifying high-impact parameters, we conduct an ablation study on high-impact parameters. Specifically, we remove the high-rank and low-rank modules with the highest IGU-LoRA scores from different layers of the Qwen2.5-0.5B model and evaluate the performance drop on the Boolq and GSM8K datasets. As shown in Table 8, removing the high-rank modules from the K module in Layer 3 (L3_K) and the V module in Layer 10 (L10_V) results in a performance drop of 1.30 and 1.33 points on Boolq, respectively. Similarly, removing the high-rank modules from the Q module in Layer 22 (L22_Q) and the K module in Layer 17 (L17_K) results in performance drops of 1.80 and 1.73 points on GSM8K, respectively. In contrast, removing the low-rank modules from the K module in Layer 1 (L1_K) and the V module in Layer 3 (L3_V) results in only minor performance drops of 0.05 and 0.10 points on Boolq, respectively. The same trend is observed on GSM8K when removing the low-rank modules from the Q module in Layer 8 (L8_Q) and the K module in Layer 6 (L6_K), resulting in performance drops of 0.11 and 0.15 points, respectively. These results demonstrate that IGU-LoRA effectively identifies high-impact parameters, as their removal leads to significant performance degradation compared to low-impact parameters.

Table 8: Ablation study on the impact of removing high-rank and low-rank modules from different layers on Qwen2.5-0.5B model performance. The numbers in parentheses indicate the performance drop compared to the model with no modules removed. The left table and the right table represent results on Boolq and GSM8K, respectively.

	Module Removed	Rank	Boolq
1	L3_K	10	81.15 (-1.30)
2	L10_V	10	81.12 (-1.33)
3	L3_K / L10_V	10 / 10	80.44 (-2.01)
4	L1_K	5	82.40 (-0.05)
5	L3_V	5	82.35 (-0.10)
6	L1_K / L3_V	5 / 5	82.30 (-0.15)
7	-	-	**82.45**

	Module Removed	Rank	GSM8K
1	L22_Q	12	32.35 (-1.80)
2	L17_K	11	32.42 (-1.73)
3	L22_Q / L17_K	12 / 11	31.15 (-3.00)
4	L8_Q	6	34.05 (-0.11)
5	L6_K	6	**34.01 (-0.15)**
6	L8_Q / L6_K	6 / 6	33.84 (-0.32)
7	-	-	**34.16**

E Generalization supplementary experiments

To further validate the generalization performance of IGU-LoRA, we conduct additional experiments on the MMLU benchmark using the Llama2-7B model. As shown in Table 9, IGU-LoRA achieves an average accuracy of 51.07%, which is very close to the full fine-tuning method (51.54%) and outperforms LoRA (49.94%). Notably, IGU-LoRA demonstrates superior performance in Science, Technology, Engineering, and Mathematics (STEM) and Social Science subjects, achieving accuracies of 41.71% and 58.12%, respectively. These results further confirm the effectiveness of IGU-LoRA in enhancing the generalization capabilities of fine-tuned models across diverse subject areas.

Table 9: The generalization performance of fine-tuning the Llama2-7B model on the MMLU benchmark using different methods, reporting the average results over 5 random seeds.

Method	**Humanities**	**STEM**	**Social.**	**Other**	**Avg.**
Full FT	**49.91**	41.70	57.53	57.02	**51.54**
LoRA	46.15	40.84	56.63	56.23	49.94
IGU-LoRA	47.33	**41.71**	**58.12**	**57.10**	51.07

F Multimodal benchmark supplementary experiments

To further demonstrate the effectiveness of IGU-LoRA in multimodal tasks, we conduct additional experiments on the VQAv2, GAQ, NVLR^2 and COCO Captioning datasets using the VL-BART (Su et al., 2019). As shown in Table 10, IGU-LoRA achieves an average score of 77.47, outperforming

LoRA (74.31) and DoRA (77.40), and closely approaching the performance of full fine-tuning (77.35). These results further validate the capability of IGU-LoRA to effectively adapt multimodal models while maintaining high performance across different tasks.

Table 10: Performance comparison of different fine-tuning methods on the VQA, GAQ, NVLR^2 and COCO datasets using the VL-BART model. The results are averaged over 5 random seeds.

Method	**VQAv2**	**GAQ**	**NVLR^2**	**COCO Captioning**	Avg.
Full FT	66.91	56.72	73.71	112.04	77.35
LoRA	64.32	54.10	71.25	109.56	74.31
DoRA	65.81	54.71	73.14	115.93	77.40
IGU-LoRA	65.78	55.32	73.42	115.36	**77.47**

G THE VERIFICATION OF THE I.I.D./LOCAL STATIONARITY APPROXIMATION IN THEOREM 2.

To validate the i.i.d. / local stationarity approximation used in Theorem 2, we conduct an empirical analysis of the importance score statistics during the fine-tuning process. Specifically, we monitor several representative modules (e.g., the L16_Q module for the 16-th layer's Q component and the L5_K module for the 5-th layer's K component) across multiple training iterations on the BoolQ dataset. We observe that, after the initial epochs, the mean and variance of y_t quickly stabilize within a narrow range, and the first-order lag autocorrelation becomes very small. Correspondingly, the SNR_t curve becomes nearly flat after the burn-in period. These observations suggest that the i.i.d./local stationarity approximation is reasonable and accurate during the stage when EMA-based importance-ranking pruning is applied in practice.

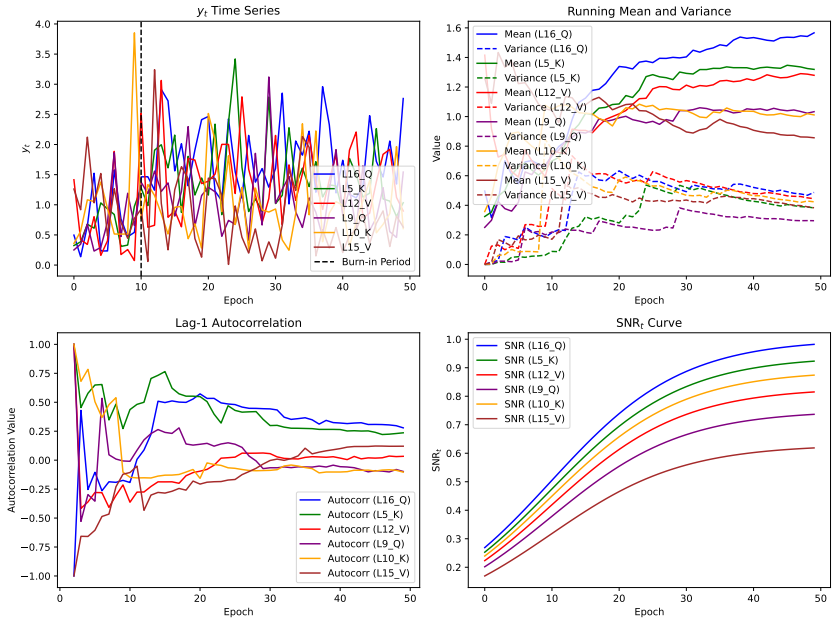

Figure 7: Empirical analysis of importance score statistics during fine-tuning. The plots show the changes in y_t, the mean and variance of y_t, the first-order lag autocorrelation, and SNR_t across training iterations for representative module parameters.

H EFFECTS OF SAMPLE ORDER AND BATCH SIZE

To investigate the effects of sample order and batch size on the performance of IGU-LoRA, we conduct experiments using the Qwen-2.5-0.5B model on the BoolQ dataset. The results are summarized as follows:

Sample Order / Random Seed. we trained with a fixed batch size using five different random seeds. These seeds control the data shuffling and the sampled integration nodes α_k. The downstream accuracy varies slightly across seeds (within Δ_{acc} absolute points, indicating a small change), which demonstrates that the sample order has high stability on the results.

Batch Size. We further vary the batch size (e.g., 2, 4, 8, 16, 32) while keeping all other hyperparameters fixed. The resulting test accuracy again shows only minor variation. This proves that batch size does not have a significant impact on the results. The detailed results are presented in Table 11.

Table 11: Effect of Batch Size on BoolQ Accuracy across Different Random Seeds

Batch Size	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
2	82.46	82.47	82.45	82.46	82.45
4	82.45	82.46	82.44	82.45	82.44
8	82.44	82.45	82.43	82.44	82.43
16	82.45	82.46	82.44	82.45	82.44
32	82.40	82.41	82.39	82.40	82.39

I Datasets and Metrics

I.1 GLUE Benchmark Tasks

Single-sentence Classification Tasks. (1) *CoLA (Corpus of Linguistic Acceptability)*: Determine whether a sentence adheres to grammatical rules (binary classification). (2) *SST-2 (Stanford Sentiment Treebank)*: Movie review sentiment analysis (positive/negative binary classification).

Sentence-pair Classification Tasks. (1) *MRPC (Microsoft Research Paraphrase Corpus)*: Determine whether two sentences are semantically equivalent (binary classification). (2) *QQP (Quora Question Pairs)*: Determine whether two Quora questions are semantically identical (binary classification). (3) *RTE (Recognizing Textual Entailment)*: Determine whether a sentence pair entails a relationship (three-class classification: entailment/contradiction/neutral).

Similarity and Regression Task. *STS-B (Semantic Textual Similarity Benchmark)*: Calculate the semantic similarity between two sentences (continuous value from 1 to 5).

Question-answering Task. *QNLI (Question-answering NLI).* Determine whether a sentence contains the answer to a given question (binary classification).

Natural Language Inference Task. *MNLI (Multi-Genre Natural Language Inference).* Large-scale cross-domain textual entailment classification (three-class classification).

I.2 Mathematical and Common-sense Reasoning Tasks

Mathematical Reasoning Tasks. (1) *AQuA (Algebra question answering)*: Derive the correct answer from a given algebraic problem (multiple-choice) and generate the corresponding solution process (Rationales). (2) *GSM8K (Grade school math 8K)*: Perform multi-step reasoning on mathematical problems described in natural language.

Common-Sense Reasoning Tasks. (1) *BoolQ (Boolean questions).* Determine whether the answer to a given question, based on the provided paragraph, is "Yes" (True) or "No" (False). (2) *ARC-e (AI2 reasoning challenge - easy)*: Select the most reasonable answer from a given set of scientific questions (Multiple-choice question). (3) *ARC-c (AI2 reasoning challenge - challenge)*: Combine multi-step reasoning and cross-domain knowledge to provide answers. (4) *COPA (Choice of plausible alternatives).* Select the most plausible cause or effect for a given premise from two provided alternatives. The task requires understanding of causal relationships and commonsense reasoning in everyday scenarios.

I.3 MultiModal Benchmark Tasks

Visual Question Answering Tasks. (1) *VQAv2 (Visual Question Answering v2).* Given an image and a related question, select the most appropriate answer from multiple choices. (2) *GAQ (Generalized*

Question Answering). This task extends VQA to a more generalized setting, where the model is asked to answer a wider range of questions based on visual context.

Visual-Linguistic Reasoning Task. (1) *NLVR2 (Natural Language for Visual Reasoning 2).* Given a pair of images and a natural language statement, determine whether the statement accurately describes the relationship between the two images.

Image Captioning Task. (1) *COCO Captioning.* Generate descriptive captions for images in the COCO dataset, evaluating the model's ability to understand and describe visual content accurately.

Table 12: Summary of the benchmark datasets.

Datasets	# train	# dev	# test	Type	Metrics
Common-Sense reasoning tasks					
BoolQ	9427	-	3270	Common-Sense reasoning	Acc
ARC-e	2251	570	2376	Common-Sense reasoning	Acc
ARC-c	1119	299	1172	Common-Sense reasoning	Acc
COPA	400	100	500	Common-Sense reasoning	Acc
Mathematical reasoning tasks					
AQuA	97467	254	254	Mathematical reasoning	Acc
GSM8K	7473	-	1319	Mathematical reasoning	Acc
GLUE benchmark tasks					
SST-2	67k	872	1.8k	Sentiment	Acc
MNLI	393k	20k	20k	NLU	Acc
QQP	364k	40k	391k	Paraphrase	Acc-F1
MRPC	3.7k	408	107k	Paraphrase	Acc-F1
RTE	2.5k	176	3k	NLU	Acc
QNLI	108k	5.7k	5.7k	QA/NLI	Acc
CoLA	8.5k	1k	1k	Acceptability	Mcc
STS-B	7k	1.5k	1.4k	Similarity	Corr

I.4 Dataset Statistics

In our experiments, we compare performance across multiple tasks, including the GLUE benchmark, which consists of eight datasets: CoLA, SST-2, MRPC, QQP, STS-B, MNLI, QNLI, and RTE; three common-sense reasoning tasks (BoolQ, ARC-e, and ARC-c); and two mathematical reasoning tasks (AQuA and GSM8K). The dataset statistics are presented in Table 12.

I.5 Evaluation Metrics

As shown in Table 12, we strictly follow the official settings of GLUE and use the same metrics as Wang et al. (2018). For MNLI, we report the average of the accuracy scores on the matched and mismatched test sets. For MRPC and QQP, we report Acc-F1, the average accuracy, and F1 scores. For STS-B, we report Corr, which denotes the average of the Pearson and Spearman correlation coefficients. For CoLA, we report Mcc, which is the Matthews correlation. For all other tasks, we report accuracy (Acc). Since the common sense and math reasoning tasks usually come with a definite answer choice, we will directly consider the correctness of the final answers. Thus, we report accuracy (denoted as Acc).

J Baseline Details

• *Full fine-tuning* is the most common approach for adaptation. During fine-tuning, the model is initialized with pre-trained weights and biases, and all model parameters undergo gradient updates.

• *LoRA* (Hu et al., 2022a) is a representative parameter-efficient fine-tuning (PEFT) method. It introduces two low-rank matrices to parameterize the incremental weight updates, and only these lightweight components are updated during fine-tuning. The number of trainable parameters is

determined by the rank r and the number of inserted adaptation matrices n, allowing for fine-grained control over the adaptation budget.

• *AdaLoRA* (Zhang et al., 2023) extends the conventional LoRA framework by introducing a dynamic rank adaptation mechanism. It parameterizes the low-rank adapters using singular value decomposition (SVD), and evaluates the importance of each parameter based on the magnitude of its corresponding singular value. This importance score then guides a progressive rank pruning process, allowing the model to dynamically reallocate its limited parameter budget to more critical layers or modules.

• *DoRA* (Liu et al., 2024b) enhances the learning capacity and adaptability of pretrained models by decoupling weight matrices into two distinct components: magnitude and direction. The key idea is to keep the magnitude fixed and apply LoRA-style low-rank updates only to the directional component. This separation allows for more expressive and geometry-aware adaptation while preserving the norm of the original weights, which helps stabilize training and maintain alignment with the pretrained model. Since only the direction is modified, DoRA introduces no additional inference overhead, making it efficient and scalable for deployment.

• *AutoLoRA* (Xu et al., 2023) is a meta-learning-based fine-tuning approach designed to automatically determine the optimal rank for each layer in Low-Rank Adaptation (LoRA). It introduces a learnable selection variable for each rank-1 matrix and dynamically adjusts these variables using a meta-learning strategy. By jointly optimizing the rank configuration along with the LoRA parameters, AutoLoRA significantly improves fine-tuning efficiency and overall performance.

• *Adapter* (Houlsby et al., 2019) inserts lightweight bottleneck modules between each layer of the pretrained model, updating only these newly introduced modules during fine-tuning while keeping the original model parameters frozen.

• *P-tuning v2* (Liu et al., 2021) is an improved prompt tuning method that inserts trainable prompt tokens at the input layer and across multiple model layers. This design increases the trainable parameters from approximately 0.01% to 0.1%-3% of the full model, while maintaining parameter efficiency. P-tuning v2 enhances optimization stability and improves performance across various tasks by integrating task-specific information deeper into the model.

• *$(IA)^3$* (Liu et al., 2022a) introduces learnable scaling vectors at key locations in the Transformer architecture, such as the keys and values in the self-attention mechanism and the intermediate activations in the feed-forward networks. These vectors are applied via element-wise multiplication to modulate the internal activations, enabling flexible control over the model's output without modifying the original model parameters.

• *SSP* (Hu et al., 2022b) leverages structural sparsity to guide the automatic search for parameter insertion locations, activating trainable parameters only in the most important substructures. This enables higher efficiency without sacrificing model performance.

• *GoRA* (He et al., 2025) leverages gradient-driven adaptive low-rank adjustment to dynamically adjust the rank of low-rank adaptation layers during training. By using gradient information, GoRA ensures that the model can allocate computational resources more efficiently, adjusting the rank based on the importance of each layer for different tasks and training stages. This method maintains computational efficiency while improving model performance, adapting the low-rank configuration to meet the specific needs of the training process.

K Additional Related Works

K.1 Dynamic Rank Allocation

Dynamic rank allocation gains increasing attention in deep learning model optimization, with various methods proposed to improve adaptability and efficiency. Several other notable approaches are introduced beyond AdaLoRA (Zhang et al., 2023) and AutoLoRA (Xu et al., 2023). LoSA (Huang et al., 2025) integrates sparsity and low-rank adaptation, dynamically adjusting both using representation mutual information and reconstruction error. PRILoRA (Benedek & Wolf, 2024) employs a heuristic strategy that linearly increases ranks from lower to higher layers, motivated by the observation that higher layers often require greater adaptability in transfer learning. ALoRA (Liu et al., 2024c) further incorporates a novel mechanism, AB-LoRA, which assesses the importance of individual LoRA

ranks and incrementally prunes redundant components, reallocating the freed budget to more critical Transformer modules. These methods provide diverse rank allocation strategies that contribute to more efficient fine-tuning of large models.

L THE USE OF LARGE LANGUAGE MODELS

During the preparation of this manuscript, large language models (LLMs) were employed in several auxiliary capacities. First, at the writing stage, LLMs were utilized to refine and translate the text, thereby enhancing the overall fluency, readability, and precision of academic expression. Second, in relation to experiments and results presentation, LLMs assisted in generating parts of the code for data visualization and figure plotting, which facilitated a more efficient presentation of research findings. Third, in surveying the research landscape and related work, LLMs provided support for literature searches, helping us to locate and summarize relevant studies in the field systematically. Finally, in the theoretical component of this work, LLMs offered auxiliary support in structuring complex proofs and verifying critical derivation steps, contributing to the clarity and rigor of our theoretical analysis. It should be emphasized that all uses of LLMs were strictly auxiliary in nature; the formulation of research questions, the design of methods, the core theoretical derivations, and the experimental analyses were all carried out independently by the authors.