LOST IN AGGREGATION: A GENERAL PERSPECTIVE ON ZERO-COST NEURAL ARCHITECTURE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Selecting neural network architectures typically requires training and comparing many candidates. Even at the same parameter scale, different connectivity patterns and operations can lead to substantial performance differences. Zero-cost neural architecture search (NAS) evaluates candidates before training and thus offers a low-cost route to architecture screening. Existing zero-cost proxies, however, use different computational forms and their common structure remains insufficiently understood. We provide a general perspective grounded in the initialization state of a network. Although existing proxies use different information, they can be understood through the local effect of the network on output changes. We show that conventional global scores lose differences between layers when aggregating layer information, while other readouts retain only local responses or intermediate features. We therefore construct architecture scores that preserve layer-wise information and aggregate it after layer-wise readout. On a strictly parameter-matched candidate subset, the proposed method provides strong architecture screening performance across several evaluations. Experiments further show that both layer-wise information and its aggregation rule materially affect zero-cost proxy rankings.

1 INTRODUCTION

Neural networks have achieved excellent results across many domains. Beyond parameter count, the network structure itself affects the outcome of training: under the same parameter budget, architectures with different connectivity patterns and operations can perform substantially differently (Zoph & Le, 2017; Liu et al., 2019; Real et al., 2019). Our empirical setting illustrates that this difference can be large even when only the architecture changes and no additional training is introduced: the accuracy range in NAS-Bench-101 reaches 30.9 percentage points (Ying et al., 2019). Designing an appropriate architecture is therefore important for the final model performance.

Architecture design is expensive because the combinatorial space induced by different connections and operations is very large. Neural architecture search automates this process by exploring a candidate space and selecting according to downstream performance, but traditional NAS repeatedly trains many candidates. For example, the NASNet search used approximately 32,400–43,200 GPU-hours (Zoph et al., 2018). ENAS reduced search cost through parameter sharing, but candidate architectures still participate in training (Pham et al., 2018). NAS-Bench-101 further demonstrates the scale of the problem: constructing the benchmark required more than 100 TPU-years (Ying et al., 2019). These limitations motivate train-free proxies that evaluate candidates without training (Cai et al., 2019; Tan et al., 2019; Wu et al., 2019; Tan & Le, 2019; Cai et al., 2020; Yu et al., 2020; Brock et al., 2018; Liu et al., 2018).

Many train-free proxies evaluate architectures at initialization. Examples include GradNorm, SNIP, GraSP, and Fisher, which use gradients or parameter sensitivity; TE-NAS, which uses spectral properties; and NASWOT, MeCo, and NEAR, which use feature representations (Abdelfattah et al., 2021; Lee et al., 2019; Wang et al., 2020; Tanaka et al., 2020; Molchanov et al., 2019; Chen et al., 2021; Mellor et al., 2021; Husistein et al., 2025; Casarin et al., 2025; Li et al., 2023; Peng et al., 2024; Dong et al., 2025; Xiang et al., 2023; 2024; Zhou et al., 2026; Han et al., 2026; Touayouch et al., 2026). These methods characterize trainability, expressivity, or parameter sensitivity at initialization and have demonstrated useful ranking ability on several NAS benchmarks (Zhou et al.,

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.

2022). Recent benchmark and survey studies further compare zero-cost proxies across tasks and expose variation in their empirical behavior (Krishnakumar et al., 2022; Li et al., 2024; Lukasik et al., 2025). However, existing zero-cost proxies exhibit inconsistent performance across search spaces and tasks, and only a few proxies generalize well across most benchmarks (Krishnakumar et al., 2022; White et al., 2021). Simple architectural baselines, such as parameter count and FLOPs, remain highly competitive with many zero-cost proxies and can outperform individual proxies on some search spaces and tasks (Krishnakumar et al., 2022; White et al., 2021; Li et al., 2024). Ranking the high-performing tail is substantially harder than ranking the full search space: prior evaluations report very low top-tail overlap for many zero-cost proxies, with several proxies performing at or below the random-selection level (Abdelfattah et al., 2021; Krishnakumar et al., 2022). Existing work usually presents and compares proxies as independent measures. We therefore study whether the timing of aggregation, namely aggregate-before-readout versus readout-before-aggregate, creates an information bottleneck in zero-cost architecture evaluation (Elsken et al., 2019; White et al., 2021; Dong & Yang, 2020; Dong et al., 2022; Klyuchnikov et al., 2020). We revisit zero-cost architecture evaluation from the information contained in the initialization state. Different methods extract different levels of information from initialization and form scores through transformations, weighting, and statistics. This viewpoint gives a general perspective for studying the structural difference between global and layer-wise information and their aggregation.

This perspective also exposes an important issue: existing methods primarily describe the response of the network as a whole, while paying less attention to how that response is distributed across layers. Two architectures may have similar global responses, while one concentrates its response in a few layers and the other distributes it over many layers. Global information can compress away this distinction. We therefore study layer-wise information and its aggregation.

We decompose the global response into parameter layers and use the response magnitude

$$E_l = \|J_l\|_F^2 \tag{1}$$

to characterize the contribution of layer l to output changes. The global response is jointly formed by all layers, while architectures may differ in how that response is distributed. We then use the layer-aware aggregation

$$S_\alpha = \sum_l E_l^\alpha \tag{2}$$

to form an architecture score. As α decreases, low-response layers receive greater relative weight. This trend agrees with the tested ranking results and is best among the evaluated aggregation choices at logarithmic aggregation,

$$S_{\log} = \sum_l \log E_l. \tag{3}$$

On a strictly parameter-matched subset, the layer-aware proxy obtains a Spearman correlation of -0.7380 and corresponding advantages in Top-K architecture hit rates.

Our contributions are:

1. We formulate zero-cost architecture evaluation as a composition of information extraction, structural representation, and statistical readout, which provides a common view of how proxies retain and compress initialization information.
2. We show that global responses lose their layer distribution and formally characterize when layer-wise scalarization preserves this hierarchical information.
3. We propose the layer-aware power aggregation $S_\alpha = \sum_l E_l^\alpha$ and prove that the relative contribution of a lower-response layer increases monotonically as α decreases. We then present logarithmic aggregation as the final architecture score.

2 Related Work

Zero-Cost Neural Architecture Search. Zero-cost neural architecture search (Zero-cost NAS) aims to evaluate neural network architectures at initialization without training candidate networks. Existing methods mainly construct proxy metrics using information such as gradients, parameter sensitivity, feature representations, or local responses. Representative methods include SNIP, GraSP,

Fisher, SynFlow, NASWOT, TE-NAS, ZiCo, NEAR, and SWAP (Lee et al., 2019; Wang et al., 2020; Molchanov et al., 2019; Tanaka et al., 2020; Mellor et al., 2021; Chen et al., 2021; Li et al., 2023; Husistein et al., 2025; Peng et al., 2024). Despite their differences in formulation, existing benchmark studies show that the performance of zero-cost proxies varies substantially across different search spaces and tasks, motivating further investigation into what information these proxies retain and how such information is transformed into the final architecture score (Krishnakumar et al., 2022; Li et al., 2024; Lukasik et al., 2025).

Layer-Wise Information and Proxy Aggregation. Existing zero-cost proxies typically compress initialization information into a global scalar, for example, through norms, traces, spectral statistics, or other summary statistics (Abdelfattah et al., 2021; Lee et al., 2019). Although some methods utilize intermediate features or layer-wise information, such as NASWOT, NEAR, and SWAP (Mellor et al., 2021; Husistein et al., 2025; Peng et al., 2024), prior work has also explored layer-wise score aggregation. Our focus is the more specific question of *when* aggregation occurs: whether global representation is formed before readout or layer-wise scalarization is performed before aggregation.

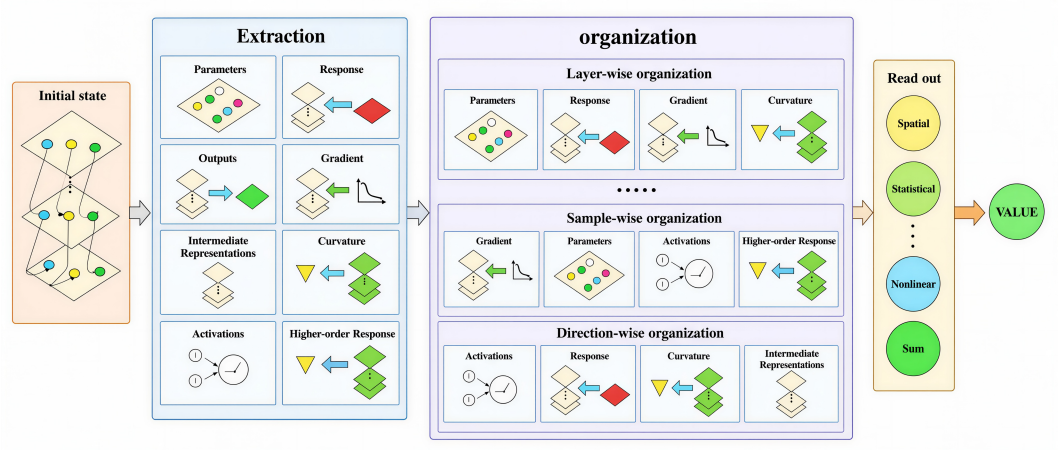


Figure 1: General perspective on zero-cost architecture evaluation. Initialization information is extracted, represented at global or layer-wise granularity, and converted to an architecture score through a statistical readout.

3 Preliminaries

3.1 Global and Layer-wise Responses at Initialization

For architecture A, fixed probe inputs $X = (x_1, \ldots, x_m)$, and initialization $\theta_{A,0}$, let $F_A(\theta; X) \in \mathbb{R}^{mC}$. The initialization Jacobian is

$$J_A = \left.\frac{\partial F_A(\theta; X)}{\partial \theta}\right|_{\theta=\theta_{A,0}},$$

and

$$F_A(\theta_{A,0} + \delta\theta_A; X) = F_A(\theta_{A,0}; X) + J_A \delta\theta_A + o(\|\delta\theta_A\|_2).$$

We write J below. Its output- and parameter-space quadratic forms are

$$K = JJ^\top, \qquad \Theta = J^\top J,$$

which have the same nonzero spectrum and rank r; if their positive eigenvalues are $\lambda_1 \geq \cdots \geq \lambda_r$, then $\mathrm{tr}(K) = \sum_i \lambda_i = \|J\|_F^2$. These quantities describe global response but do not identify its layer allocation.

Partition the complete parameter vector into B blocks, $\theta = (\theta_1, \ldots, \theta_B)$, with $J = [J_1, \ldots, J_B]$. Define

$$K_b = J_b J_b^\top, \qquad E_b = \mathrm{tr}(K_b) = \|J_b\|_F^2,$$

so $K = \sum_{b=1}^{B} K_b$ and $\sum_b E_b = \mathrm{tr}(K)$. For layer-wise scoring, select the L Conv2d/Linear weight blocks and re-index their Jacobians and responses as J_l and E_l.

4 A General Perspective

4.1 Local information at initialization

For candidate A with initialization $\theta_{A,0}$ and probe inputs $X = (x_1, \ldots, x_m)$, let $F_A(\theta_{A,0}, X)$ denote the output at initialization. The initialization state exposes outputs, intermediate representations $\{Z_l\}$, activation states $\{\mathcal{A}_l\}$, and local responses to parameter, input, or intermediate-state perturbations. Representative first and higher order quantities are

$$D_\theta F_A, \quad D_x F_A, \quad \{D_{Z_l} F_A\}, \quad D_\theta^2 F_A, \quad D_\theta \nabla_\theta \mathcal{L}.$$

We collect the information retained up to local order r in $\mathcal{I}_A^{(r)}$ and write a general zero-cost proxy as

$$S_A = \Phi_A(\mathcal{I}_A^{(r)}; X, Y), \tag{4}$$

where Y is used when a proxy requires supervision.

Representation based methods primarily use $\{Z_l\}$ or $\{\mathcal{A}_l\}$, as in NASWOT, NEAR, SWAP, MeCo, and Zen (Mellor et al., 2021; Husistein et al., 2025; Peng et al., 2024; Dong et al., 2025; Xiang et al., 2023; Lin et al., 2021). Gradient and sensitivity methods use parameter output responses such as $D_\theta F_A$; GradNorm, SNIP, Fisher, NTK, and related methods differ in how they select, weight, and aggregate this information (Abdelfattah et al., 2021; Lee et al., 2019; Molchanov et al., 2019; Jacot et al., 2018). Methods that use curvature add higher order local structure, as in GraSP (Wang et al., 2020). ZiCo retains the sample dimension: for probe sample i, with local response $J^{(i)}$ and output direction u_i, it forms

$$g_i = (J^{(i)})^\top u_i,$$

and then computes statistics across i (Li et al., 2023). The framework therefore distinguishes information by what is retained from initialization and by when it is compressed.

4.2 General representation

Parameter-induced changes have output-direction, parameter-direction, sample, and layer-distribution dimensions. With $J = [J_1, \ldots, J_L]$, define

$$K_l = J_l J_l^\top, \qquad K = \sum_{l=1}^{L} K_l.$$

Retaining $\{K_l\}$ preserves hierarchy, whereas K compresses it. Direction selection, sample aggregation, and statistical readout applied before or after this compression produce different scores: for a fixed output direction u, $\|J^\top u\|_2^2 = u^\top J J^\top u$, while $\mathrm{tr}(JJ^\top) = \|J\|_F^2$ summarizes all output directions. Parameter-dependent weights modify parameter-direction contributions, and per-sample statistics preserve information that early sample merging removes. Finally, trace, eigenvalue, effective-rank, power, and logarithmic statistics provide distinct readouts. In particular,

$$\Phi\left(\sum_l K_l\right) \neq \Psi\left(\{\Phi_l(K_l)\}_l\right) \tag{5}$$

for general nonlinear readouts, although trace is linear. Layer-aware scoring therefore retains information before cross-layer aggregation. Figure 1 summarizes this extraction, representation, and readout pipeline.

For the global local-response matrix

$$J = \left.\frac{\partial F_A(\theta; X)}{\partial \theta}\right|_{\theta=\theta_{A,0}},$$

the global output-space representation is

$$K = JJ^\top.$$

This merges first-order information from all layers into one output-space object, which facilitates global statistics and spectral analysis but no longer explicitly preserves layer identity.

5 LAYER-WISE SCORING

5.1 GLOBAL AGGREGATION AND LOSS OF LAYER INFORMATION

Partition $J = [J_1, \dots, J_L]$ and let $K_l = J_l J_l^\top$. Global aggregation gives

$$K = JJ^\top = \sum_{l=1}^{L} K_l,$$

so the map $\{K_l\}_l \mapsto K$ sums over layers. It is therefore generally impossible to recover all layer responses from K alone. For example,

$$J^{(a)} = \begin{bmatrix} \sqrt{3/2} & \sqrt{1/2} \end{bmatrix}, \qquad J^{(b)} = [1 \quad 1] \tag{6}$$

have the same $JJ^\top = 2$ but layer responses $(3/2, 1/2)$ and $(1, 1)$, respectively. This is the first theoretical result.

Proposition 1 (global non-identifiability). In general, no function depending only on $JJ^\top$ can recover $(E_1, \dots, E_L)$ for every J. Consequently, global trace and spectral statistics retain total response but not its allocation across layers. A directional readout likewise gives only $\|J_l^\top u\|_2^2 = u^\top K_l u$, whereas $E_l = \mathrm{tr}(K_l)$ summarizes all output directions. *Proof.* The two matrices displayed above have the same global representation, $K^{(a)} = K^{(b)} = 2$, but different layer responses. Any function of K alone must assign the same output to these two cases, so it cannot recover both response vectors. The contradiction is structural: summation preserves the total trace, but removes the decomposition into summands. The same argument applies to any statistic computed solely from the global trace or spectrum, because those statistics are identical whenever the global matrix is identical. A directional readout is also incomplete, since one vector u observes only one quadratic projection of each K_l, whereas the trace aggregates all output directions. See Appendix A.1 for the formal contradiction and its implications for power and logarithmic readouts.

5.2 LAYER-WISE POWER AGGREGATION

Use the layer-wise scalar response

$$E_l = \mathrm{tr}(K_l) = \|J_l\|_F^2, \qquad M_E = \sum_{l=1}^{L} E_l, \qquad q_l = E_l / M_E. \tag{7}$$

Thus $E_l = M_E q_l$: M_E is total response and q is its layer distribution. For $\alpha \neq 0$,

$$M_\alpha(E) = \left(L^{-1} \sum_l E_l^\alpha \right)^{1/\alpha} = M_E L^{-1/\alpha} \left(\sum_l q_l^\alpha \right)^{1/\alpha}. \tag{8}$$

Proposition 2 (monotone relation between α and the relative contribution of lower-response layers). For $E_i > E_j > 0$, the relative contribution of the lower-response layer is $(E_j/E_i)^\alpha$, which strictly increases as α decreases. *Proof.* Let $\gamma = E_j/E_i \in (0, 1)$. Then $\frac{d}{d\alpha}\gamma^\alpha = \gamma^\alpha \log \gamma < 0$, so decreasing α increases γ^α. In the power sum, this ratio compares the two layer contributions directly; hence the effect is independent of the absolute scale of E_i and E_j and depends only on their relative imbalance. As $\alpha \to 0$, the ratio approaches one, so the aggregation progressively exposes layers that contribute less under ordinary summation. The detailed derivative argument is given in Appendix A.2.

Proposition 3 (balance under fixed total response). For fixed M_E and $0 < \alpha < 1$, M_α is maximized by $E_l = M_E/L$ for every layer and minimized, in the closure $E_l \geq 0$, by concentrating all response in one layer, giving

$$\frac{M_E}{L^{1/\alpha}} \leq M_\alpha(E) \leq \frac{M_E}{L}. \tag{9}$$

Proof. Because x^α is strictly concave on $x > 0$, transferring a small amount of response from a larger layer to a smaller one increases $\sum_l E_l^\alpha$ while preserving M_E. Repeating this balancing operation

yields the uniform vector and the upper bound. At the opposite boundary, concentrating all response in one layer gives the lower value because the remaining terms vanish. Thus the generalized mean distinguishes not only response scale but also its distribution: with fixed total response, a more balanced allocation produces a larger value when $0 < \alpha < 1$. A full transfer calculation and the boundary argument appear in Appendix A.2.

5.3 Logarithmic aggregation and implemented score

The normalized $\alpha \to 0$ limit is the geometric mean. Under $E_l > 0$,

Proposition 4 (logarithmic/KL decomposition). The logarithmic score satisfies

$$M_0(E) = \left(\prod_{l=1}^{L} E_l \right)^{1/L}, \qquad S_{\log} = \sum_{l=1}^{L} \log E_l = L \log M_0(E), \tag{10}$$

and

$$\log M_0(E) = \log(M_E/L) - D_{\mathrm{KL}}(q^{\mathrm{uni}} \| q), \qquad q_l^{\mathrm{uni}} = 1/L. \tag{11}$$

Proof. Since $S_{\log} = L \log M_0(E)$, the first identity follows directly from the definition of the geometric mean. Substituting $E_l = M_E q_l$ gives

$$\frac{1}{L} \sum_l \log E_l = \log(M_E/L) + \frac{1}{L} \sum_l \log(L q_l).$$

The second term equals $-D_{\mathrm{KL}}(q^{\mathrm{uni}} \| q)$ because $q_l^{\mathrm{uni}} = 1/L$. This decomposition separates two effects: M_E measures the total response scale, while the KL term decreases the score when response is unevenly allocated across layers. In particular, with M_E fixed, the score is largest for the uniform distribution and becomes smaller as mass concentrates on fewer layers. See Appendix A.3 for the limit and full decomposition. Thus logarithmic aggregation combines total response with a penalty for uneven layer allocation.

5.4 Design of the layer-aware score

For an architecture A, let $\mathcal{L}_A = \{l : E_l(A) > \tau\}$ denote the active response blocks, where $\tau = 10^{-30}$, and let $L_A = |\mathcal{L}_A|$. Blocks with $E_l(A) \le \tau$ are excluded before layer-wise aggregation and layer statistics. The implemented logarithmic score is

$$S_{\log,\tau}(A) = \sum_{l \in \mathcal{L}_A} \log E_l(A), \qquad \tau = 10^{-30}, \tag{12}$$

where the logarithm is natural. For the implementation, M_E and q_l are evaluated over $\mathcal{L}_A$, so the retained responses satisfy $\sum_{l \in \mathcal{L}_A} q_l = 1$. The two computation paths are

$$\{J_l\} \to \{K_l\} \to \sum_l K_l, \qquad \{J_l\} \to \{E_l\} \to \{\mathcal{L}_A\} \to \sum_{l \in \mathcal{L}_A} \log E_l,$$

which differ in whether layer identity is removed before nonlinear readout. Appendix A.4 records the separate numerical-stabilization bound for the theoretical ϵ-regularized form.

6 Experiments

We evaluate the proposed representation and readout on a strictly parameter-matched NAS-Bench-101 subset (Ying et al., 2019). The parameter count is 5,793,546 and the final set contains 3000 architectures. For each architecture, the ranking target is the mean of the three independent training repeats at final epoch 108, $\bar{v}_A = \frac{1}{3} \sum_{r=1}^{3} v_{A,r}^{(108)}$. All architectures use the same initialization seed, CIFAR-10 probe, benchmark construction, and stem setting. The probe size is 100 and networks are in training mode. Proxies are computed at initialization without training candidates. We compare rank correlation, aggregation exponents, layer response distributions, and Top-K screening.

Table 1: Ranking results on strictly parameter-matched architectures.

No.	Proxy	Representation	Spearman (ρ)	Kendall (τ_b)
1	$\boldsymbol{S_{\log,\tau} = \sum_{l\in\mathcal{L}_A} \log E_l}$	Layer-wise response	$\mathbf{-0.7380}$	$\mathbf{-0.5420}$
2	$\sum_l E_l^{1/4}$	Layer-wise response	-0.7104	-0.5165
3	GradNorm-sum	Parameter response	-0.6910	-0.4990
4	SNIP	Parameter response	-0.6893	-0.4982
5	$\sum_l E_l^{1/2}$	Layer-wise response	-0.6827	-0.4920
6	GradNorm-(ℓ_2)	Parameter response	-0.6560	-0.4683
7	$\sum_l E_l$	Layer-wise response	-0.6558	-0.4688
8	Raw response Trace	Global response	-0.6557	-0.4688
9	Task-weighted Trace	Global response	-0.6474	-0.4618
10	$\max_l E_l$	Layer-wise response	-0.6471	-0.4616
11	$\sum_l E_l^2$	Layer-wise response	-0.6454	-0.4591
12	$\min_l E_l$	Layer-wise response	-0.6350	-0.4500
13	$\lambda_{\max}$	Global response	-0.6229	-0.4413
14	LogDet	Global response	-0.6081	-0.4287
15	Zen	Input response	-0.5811	-0.4063
16	$\lambda_{\min}$	Global response	-0.5197	-0.3595
17	SynFlow	Parameter response	+0.4165	+0.3146
18	GraSP	Parameter response	+0.3635	+0.2596
19	NEAR: ER-pre-mean	Layer-wise representation	+0.2900	+0.1974
20	MeCo	Layer-wise representation	+0.2110	+0.1406
21	SWAP	Sample-level representation	+0.1780	+0.1178
22	Participation Ratio	Global response	-0.1231	-0.0833
23	ZiCo	Per-sample parameter response	-0.1193	-0.0803
24	NASWOT	Layer-wise activation representation	-0.0123	-0.0065

6.1 Ranking on strictly parameter-matched architectures

Across all 3000 candidates, $S_{\log,\tau}$ has Spearman correlation -0.7380 and Kendall correlation -0.5420, the largest absolute correlations among the methods in Table 1; the next strongest layer aggregation, $\sum_l E_l^{1/4}$, reaches -0.7104 and -0.5165. Global indicators give a different ordering: $\lambda_{\max}$, $\mathrm{tr}(JJ^\top)$, and $\mathrm{tr}(GJJ^\top)$ have Spearman correlations -0.6229, -0.6557, and -0.6474, respectively. The near agreement between the two unweighted traces isolates the scale of the response as a stable global signal, while their gap from the layer-aware scores shows that the discarded layer allocation supplies additional ordering information. Because the correlations are negative, lower proxy values correspond to the preferred end of the architecture ranking.

6.2 Ranking under different aggregation exponents

To isolate aggregation, we fix the layer responses E_l and vary α. Because all rows in Table 2 use the same measurements, their ranking differences arise from the readout. As α decreases, the gap between the contributions of large and small responses narrows: high-response layers remain influential, but weak layers gain relative weight. The score thus reflects both total response and its distribution across stages. The observed ranking trend across tested choices agrees with the theoretical monotone relation.

Table 2: Ranking under different aggregation exponents.

Aggregation	α	Spearman (ρ)
$\sum_l E_l$	1	-0.6558
$\sum_l E_l^{1/2}$	$1/2$	-0.6827
$\sum_l E_l^{1/4}$	$1/4$	-0.7104
$\sum_l \log E_l$	$\alpha \to 0$	$\mathbf{-0.7380}$

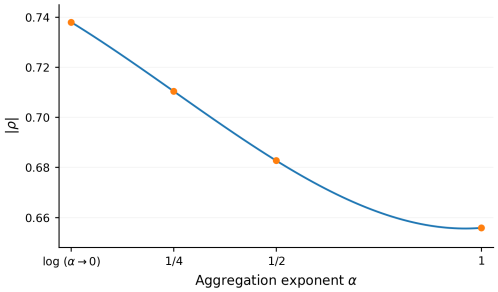


Figure 2: Ranking correlation as a function of the aggregation exponent. Markers are the values in Table 2; the line interpolates the reported points.

6.3 Global and layer-wise responses

Partition the complete response by parameter block, and let $\mathcal{W}$ be the set of Conv2d/Linear weight blocks with total response $S_{\text{weight}} = \sum_{l \in \mathcal{W}} \|J_l\|_F^2$. On 3000 candidates these blocks contribute $S_{\text{weight}}/\operatorname{tr}(JJ^\top) \approx 0.988$ of the complete raw response, so replacing the complete parameter set by the dominant weight blocks has little effect on the ordinary trace, whereas a layer-aware readout changes how that response is interpreted.

A task-weighted trace $\operatorname{tr}(GJJ^\top)$, with sample-wise output metric $G_i = \operatorname{diag}(p_i) - p_i p_i^\top$ for softmax output p_i, gives $\rho(S_{\text{task}}, Acc) = -0.6474$, slightly weaker than the unweighted summaries in Table 1 (-0.6558 and -0.6557). Output weighting changes emphasis across output directions, while layer-wise scoring changes the stage at which information is aggregated: once the layer contributions have been collapsed into $JJ^\top$, direction weighting cannot reconstruct which stages produced that response, whereas the layer-aware readout preserves this structural coordinate before aggregation.

6.4 Layer-wise results

Table 3 reports the layer-wise statistics. For the active blocks defined above, the code computes effective rank as $\exp(-\sum_l q_l \log q_l)$, kappa as $\max_l E_l / \min_l E_l$, participation as $\sum_l q_l^2$, the Top-1 response fraction as $\max_l q_l$, and the Top-2 response fraction as $(E_{(1)} + E_{(2)})/\sum_l E_l$, where $E_{(1)}$ and $E_{(2)}$ are the two largest active responses.

Table 3: Layer-wise results. Effective rank, kappa, participation, and response fractions use the code definitions stated in the text.

No.	Metric	Spearman (ρ)	Kendall (τ_b)
1	$S_{\log,\tau} = \sum_{l \in \mathcal{L}_A} \log E_l$	-0.7380	-0.5420
2	$\sum_l E_l^{1/4}$	-0.7104	-0.5165
3	$\sum_l E_l^{1/2}$	-0.6827	-0.4920
4	$\sum_l E_l$	-0.6558	-0.4688
5	$-\operatorname{Var}_l(\log E_l)$	$\mathbf{+0.3562}$	$\mathbf{+0.2436}$
6	$S_{\log q}$	+0.3442	+0.2342
7	$\min_l q_l$	+0.3071	+0.2102
8	Layer effective rank $\exp\left(-\sum_l q_l \log q_l\right)$	+0.2433	+0.1650
9	Layer entropy	+0.2432	+0.1640
10	$-\operatorname{CV}(E)$	+0.1698	+0.1144
11	Layer kappa $\max_l E_l / \min_l E_l$	-0.3000	-0.2055
12	Top-2 response fraction $(E_{(1)} + E_{(2)})/\sum_l E_l$	-0.2219	-0.1502
13	Layer participation $\sum_l q_l^2$	-0.1746	-0.1180
14	Top-1 response fraction $\max_l q_l$	-0.1512	-0.1019

The layer-wise aggregation metrics are negatively correlated with validation performance, whereas normalized distribution metrics are positively correlated (Table 3). The distinction is informative because it separates two coordinates of the same response: $S_{\log} = L \log M_E + \sum_l \log q_l$ rewards overall scale while the normalized term evaluates how that scale is allocated. In the table, power aggregation retains the useful scale signal, while $-\operatorname{Var}_l(\log E_l)$, $S_{\log q}$, and $\min_l q_l$ expose an additional distribution signal. Their positive correlations indicate that, within this candidate set, architectures with less concentrated response tend to have higher validation accuracy. Each network contains 47 Conv2d/Linear weight blocks. The mean layer effective rank is about 5.54, mean layer entropy 1.69, mean layer participation 0.354, median layer kappa 3.17×10^4, and mean $\min_l q_l$ 3.39×10^{-5}. These values describe a heterogeneous response profile in which the allocation across stages remains measurable, giving the scalar layer distribution an independent role in ranking.

6.5 Top-K architecture screening

Let $\mathcal{T}_K$ denote the K architectures with highest validation performance and let $\widehat{\mathcal{T}}_K^{(s)}$ be the Top-K set selected by proxy s. Define

$$\text{Overlap@}\,K(s) = \frac{|\widehat{\mathcal{T}}_K^{(s)} \cap \mathcal{T}_K|}{K}.$$

For negatively correlated proxies we select from low to high scores; for positively correlated proxies we select from high to low scores.

Table 4: Top-K screening results on 3000 architectures.

No.	Proxy	Overlap@50	Overlap@100
1	$\boldsymbol{S_{\log,\tau}}$	**0.160**	**0.270**
2	$\sum_l E_l^{1/4}$	0.120	0.260
3	SNIP	0.140	0.240
4	GradNorm-sum	0.120	0.230
5	$\sum_l E_l^{1/2}$	0.100	0.230
6	Task-weighted Trace	0.080	0.200
7	$\lambda_{\max}$	0.080	0.200
8	Raw response Trace	0.060	0.210
9	$\sum_l E_l$	0.080	0.200
10	Zen	0.000	0.020

On 3000 candidates, $S_{\log,\tau}$ achieves Overlap@50 $= 0.160$ and Overlap@100 $= 0.270$, the highest values in Table 4. The corresponding power score $\sum_l E_l^{1/4}$ reaches 0.120 and 0.260; the larger gain at the 50 architecture budget suggests that logarithmic aggregation separates the very top candidates more effectively, while its smaller advantage at 100 indicates that both readouts retain much of the same broad signal. Combined with the exponent sweep, this screening behavior ties the discrete selection result to the continuous theory: changing the aggregation rule changes the relative cost of weak layers and consequently changes which architectures enter the selected set.

7 Conclusion

This paper starts from the network initialization state, provides a general perspective on different information representations in zero-cost architecture evaluation, and further investigates the difference between overall responses and layer-wise responses. We prove that overall aggregation cannot preserve the distributional information of responses across different layers. We construct power aggregation and its logarithmic form, and decompose architecture responses into overall magnitude and inter-layer distribution. Experimental results show that different aggregation methods lead to different architecture rankings, and logarithmic aggregation achieves higher scores on a candidate set with strict parameter matching. Layer-wise responses provide additional layer-level information for zero-cost architecture evaluation, and the aggregation method determines how this information affects the final architecture score.

AI USE STATEMENT

Throughout the research process, ChatGPT was used to support research ideation, including formulating and refining hypotheses, brainstorming, and discussing research methodology and experimental design. It also assisted with theoretical reasoning and proof development, implementation and debugging of the proposed method, and the analysis and interpretation of experimental results. Beyond these research tasks, it was used for translation, drafting and language polishing, literature search and summarization, figure generation or modification, and reference formatting.

All AI-assisted work was reviewed by the authors. AI-assisted code was tested and verified, and relevant literature, factual claims, and technical content were independently checked. The authors take full responsibility for the final content of this paper, including all text, claims, analyses, code, figures, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, model architecture, training procedure, datasets, and evaluation protocols in the main paper and appendix. Hyperparameter settings and additional implementation details are also provided to facilitate reproduction of the reported results. The source code, experimental configurations, and datasets used in our experiments are provided as supplementary material. Complete proofs and additional theoretical details are provided in the appendix.

REFERENCES

Mohamed S. Abdelfattah, Abhinav Mehrotra, Lukasz Dudziak, and Nicholas D. Lane. Zero-cost proxies for lightweight nas. In *International Conference on Learning Representations*, 2021.

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. Smash: One-shot model architecture search through hypernetworks. In *International Conference on Learning Representations*, 2018.

Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

Sofia Casarin, Sergio Escalera, and Oswald Lanz. L-swag: Layer-sample wise activation with gradients information for zero-shot nas on vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4441–4451, 2025.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021.

Peijie Dong, Lujun Li, Zhenheng Tang, Xiang Liu, Zimian Wei, Qiang Wang, and Xiaowen Chu. Parzc: Parametric zero-cost proxies for efficient nas. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 16327–16335, 2025. doi: 10.1609/aaai.v39i15.33793.

Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations*, 2020.

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3634–3646, 2022. doi: 10.1109/TPAMI.2021.3054824.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

Chan Sik Han, Sun Woo Jeong, and Keon Myung Lee. Fab: Function-aware fisher information proxy for training-free neural architecture search, 2026. SSRN preprint.

Raphael T. Husistein, Markus Reiher, and Marco Eckhoff. Near: A training-free pre-estimator of machine learning model performance. In *International Conference on Learning Representations*, 2025.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, Alexander Filippov, and Evgeny Burnaev. Nas-bench-nlp: Neural architecture search benchmark for natural language processing. *arXiv preprint arXiv:2006.07116*, 2020.

Arjun Krishnakumar, Colin White, Arber Zela, Renbo Tu, Mahmoud Safari, and Frank Hutter. Nasbench-suite-zero: Accelerating research on zero cost proxies. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, pp. 28037–28051, 2022. doi: 10.52202/068431-2033.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Guihong Li, Yuedong Yang, Kartikeya Bhardwaj, and Radu Marculescu. Zico: Zero-shot nas via inverse coefficient of variation on gradients. *arXiv preprint arXiv:2301.11300*, 2023.

Guihong Li, Duc Hoang, Kartikeya Bhardwaj, Ming Lin, Zhangyang Wang, and Radu Marculescu. Zero-shot neural architecture search: Challenges, solutions, and opportunities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7618–7635, 2024. doi: 10.1109/TPAMI.2024.3395423.

Ming Lin, Pichao Wang, Zhenhong Sun, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Zen-nas: A zero-shot nas for high-performance image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 347–356, 2021.

Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *European Conference on Computer Vision*, pp. 19–34, 2018.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

Jovita Lukasik, Michael Moeller, and Margret Keuper. An evaluation of zero-cost proxies—from neural architecture performance prediction to model robustness. *International Journal of Computer Vision*, 133:2635–2652, 2025. doi: 10.1007/s11263-024-02265-7.

Joe Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. In *International Conference on Machine Learning*, volume 139, pp. 7588–7598, 2021.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Yameng Peng, Andy Song, Haytham M. Fayek, Vic Ciesielski, and Xiaojun Chang. Swap-nas: Sample-wise activation patterns for ultra-fast nas. In *International Conference on Learning Representations*, 2024.

Hieu Pham, Melody Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *International Conference on Learning Representations*, 2018.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, pp. 4780–4789, 2019.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114, 2019.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, 2020.

Hassan Touayouch, Rabie Najem, and Mohammed Benjelloun. Zaps: Zero-cost active proxy search for neural architecture search, 2026.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

Colin White, Arber Zela, Binxin Ru, Yang Liu, and Frank Hutter. How powerful are performance predictors in neural architecture search? In *Advances in Neural Information Processing Systems*, 2021.

Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10734–10742, 2019.

Lichuan Xiang, Lukasz Dudziak, Mohamed S. Abdelfattah, Thomas Chau, Nicholas D. Lane, and Hongkai Wen. Zero-cost operation scoring in differentiable architecture search. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 10453–10463, 2023. doi: 10.1609/aaai.v37i9.26243.

Lichuan Xiang, Lukasz Dudziak, Mohamed S. Abdelfattah, Abhinav Mehrotra, Nicholas D. Lane, and Hongkai Wen. Towards neural architecture search through hierarchical generative modeling. In *International Conference on Machine Learning*, volume 235, pp. 54236–54254, 2024.

Chris Ying, Aaron Klein, Esteban Real, Eric Christiansen, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, pp. 7105–7114, 2019.

Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas S. Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pp. 702–717, 2020.

Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Xing Sun, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10894–10903, 2022.

Qinqin Zhou, Fuhai Chen, Jipeng Wu, Zhiwei Chen, Zhikai Hu, and Weiwei Cai. Intrain: Intrinsic trainability for zero-cost neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20181–20190, 2026.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710, 2018.

A Theoretical Analysis of Layer-wise Response Aggregation

This appendix gives proofs and technical derivations omitted from the main text; definitions, central conclusions, and experimental interpretation are not repeated.

A.1 Proof of Proposition 1: global representation does not determine layer-wise response

Given

$$J = [J_1, \dots, J_L], \qquad K = JJ^\top = \sum_{l=1}^{L} J_l J_l^\top,$$

and

$$E_l = \operatorname{tr}(J_l J_l^\top) = \|J_l\|_F^2,$$

consider one output, two layers, and one parameter per layer. Take

$$J^{(a)} = \begin{bmatrix} \sqrt{3/2} & \sqrt{1/2} \end{bmatrix}, \qquad J^{(b)} = [1 \quad 1].$$

Then

$$J^{(a)}(J^{(a)})^\top = \frac{3}{2} + \frac{1}{2} = 2,$$

and

$$J^{(b)}(J^{(b)})^\top = 1 + 1 = 2.$$

Thus $K^{(a)} = K^{(b)}$, but

$$(E_1^{(a)}, E_2^{(a)}) = \left(\frac{3}{2}, \frac{1}{2}\right), \qquad (E_1^{(b)}, E_2^{(b)}) = (1, 1).$$

The same global representation therefore corresponds to different layer-wise distributions. If a function h based only on K uniquely recovered layer-wise response, then $K^{(a)} = K^{(b)}$ would imply

$$h(K^{(a)}) = h(K^{(b)}),$$

and hence $(E_1^{(a)}, E_2^{(a)}) = (E_1^{(b)}, E_2^{(b)})$, a contradiction. Therefore no such function exists. □

Although K determines

$$M_E = \sum_{l=1}^{L} E_l = \operatorname{tr}(K),$$

it does not uniquely determine $(E_1, \dots, E_L)$. Consequently, no global spectral statistic depending only on K can uniquely recover the complete layer-wise distribution. The counterexample also gives, for any $0 < \alpha < 1$,

$$\left(\frac{3}{2}\right)^\alpha + \left(\frac{1}{2}\right)^\alpha \neq 1^\alpha + 1^\alpha,$$

and

$$\log\frac{3}{2} + \log\frac{1}{2} = \log\frac{3}{4} \neq 0 = \log 1 + \log 1.$$

Thus $S_\alpha = \sum_l E_l^\alpha$ and $S_{\log} = \sum_l \log E_l$ also cannot be uniquely determined by K. Their difference from ordinary summation is due to the distribution term $\sum_l q_l^\alpha$.

For $E_i > E_j > 0$,

$$\frac{E_j^\alpha}{E_i^\alpha} = \left(\frac{E_j}{E_i}\right)^\alpha.$$

With $\gamma = E_j/E_i$, $0 < \gamma < 1$,

$$\frac{d}{d\alpha}\gamma^\alpha = \gamma^\alpha \log\gamma < 0.$$

A.2 LAYER-DISTRIBUTION PROPERTIES OF POWER AGGREGATION

Assume

$$E_l \geq 0, \qquad M_E = \sum_{l=1}^{L} E_l > 0,$$

and define $q_l = E_l / M_E$. Then

$$q_l \geq 0, \qquad \sum_{l=1}^{L} q_l = 1, \qquad E_l = M_E q_l,$$

and $(E_1, \ldots, E_L) \iff (M_E, q_1, \ldots, q_L)$ is an equivalent representation. For $\alpha > 0$,

$$M_\alpha(E) = \left(\frac{1}{L} \sum_{l=1}^{L} E_l^\alpha \right)^{1/\alpha}$$

and

$$\boxed{M_\alpha(E) = M_E L^{-1/\alpha} \left(\sum_{l=1}^{L} q_l^\alpha \right)^{1/\alpha}}. \tag{13}$$

For $E_i > E_j > 0$, setting $\gamma = E_j / E_i$ gives

$$\frac{E_j^\alpha}{E_i^\alpha} = \gamma^\alpha, \qquad \frac{d}{d\alpha} \gamma^\alpha = \gamma^\alpha \log \gamma < 0,$$

and hence

$$\boxed{\alpha \downarrow \implies \frac{E_j^\alpha}{E_i^\alpha} \uparrow .} \tag{14}$$

This derivative establishes the monotonicity used in Proposition 2. It also clarifies the interpretation of the exponent: power aggregation does not alter the layer responses themselves, but changes their relative influence in the scalar readout. For fixed total response $M_E = \sum_l E_l$, with $0 < \alpha < 1$, the function x^α is strictly concave. For $x > y > 0$, transfer δ from x to y:

$$x' = x - \delta, \qquad y' = y + \delta, \qquad 0 < \delta < \frac{x - y}{2}.$$

Define

$$\phi(\delta) = (x - \delta)^\alpha + (y + \delta)^\alpha.$$

Then

$$\phi'(\delta) = \alpha \left[(y + \delta)^{\alpha - 1} - (x - \delta)^{\alpha - 1} \right].$$

Since $y + \delta < x - \delta$ and $\alpha - 1 < 0$,

$$(y + \delta)^{\alpha - 1} > (x - \delta)^{\alpha - 1},$$

so

$$\phi'(\delta) > 0.$$

Thus transferring response from a larger layer to a smaller layer strictly increases aggregation. The balanced distribution maximizes the generalized mean:

$$\boxed{M_\alpha^{\max} = \frac{M_E}{L}.} \tag{15}$$

Concentration in one layer gives

$$\boxed{M_\alpha(E) = \frac{M_E}{L^{1/\alpha}}}, \tag{16}$$

and therefore

$$\boxed{\frac{M_E}{L^{1/\alpha}} \leq M_\alpha(E) \leq \frac{M_E}{L}, \qquad 0 < \alpha < 1.} \tag{17}$$

The continuous limit is

$$\boxed{M_0(E) = \exp\left(\frac{1}{L} \sum_{l=1}^{L} \log E_l \right) = \left(\prod_{l=1}^{L} E_l \right)^{1/L}}. \tag{18}$$

Since $S_\alpha = \sum_l E_l^\alpha = L M_\alpha(E)^\alpha$, S_α and M_α induce the same ordering for fixed L.

A.3 LOGARITHMIC AGGREGATION AND ITS DECOMPOSITION

For all $E_l > 0$, define

$$M_\alpha(E) = \left(\frac{1}{L} \sum_{l=1}^{L} E_l^\alpha \right)^{1/\alpha}, \qquad \alpha \neq 0.$$

Then

$$\log M_\alpha(E) = \frac{1}{\alpha} \log \left(\frac{1}{L} \sum_{l=1}^{L} e^{\alpha \log E_l} \right).$$

Moreover,

$$\frac{d}{d\alpha} \log \left(\frac{1}{L} \sum_{l=1}^{L} E_l^\alpha \right) \Bigg|_{\alpha=0} = \frac{1}{L} \sum_{l=1}^{L} \log E_l.$$

By continuous extension,

$$\boxed{\lim_{\alpha \to 0} \log M_\alpha(E) = \frac{1}{L} \sum_{l=1}^{L} \log E_l,} \tag{19}$$

so

$$\boxed{M_0(E) = \exp \left(\frac{1}{L} \sum_{l=1}^{L} \log E_l \right) = \left(\prod_{l=1}^{L} E_l \right)^{1/L}.}$$

The associated layer-wise logarithmic aggregation is

$$\boxed{S_{\log} = \sum_{l=1}^{L} \log E_l = L \log M_0(E).} \tag{20}$$

This is not the direct limit of S_α, since $\lim_{\alpha \to 0} S_\alpha = L$. The continuous limit occurs after the normalized logarithm:

$$\boxed{\lim_{\alpha \to 0} \frac{1}{\alpha} \log \left(\frac{S_\alpha}{L} \right) = \frac{1}{L} \sum_{l=1}^{L} \log E_l.} \tag{21}$$

Thus, for fixed L, $S_{\log}$ and $M_0(E)$ have exactly the same ordering.

Using $E_l = M_E q_l$ and $q_l^{\text{uni}} = 1/L$,

$$M_0(E) = M_E \left(\prod_{l=1}^{L} q_l \right)^{1/L},$$

$$\left(\prod_{l=1}^{L} q_l \right)^{1/L} = \frac{1}{L} \exp \left(-D_{\text{KL}}(q^{\text{uni}} \| q) \right),$$

and hence

$$\boxed{M_0(E) = \frac{M_E}{L} \exp \left(-D_{\text{KL}}(q^{\text{uni}} \| q) \right),} \tag{22}$$

$$\boxed{\log M_0(E) = \log \frac{M_E}{L} - D_{\text{KL}}(q^{\text{uni}} \| q),} \tag{23}$$

$$\boxed{S_{\log} = L \log \frac{M_E}{L} - L D_{\text{KL}}(q^{\text{uni}} \| q).} \tag{24}$$

Because $D_{\text{KL}}(q^{\text{uni}} \| q) \geq 0$, with equality only at $q_1 = \cdots = q_L = 1/L$, a more uneven allocation gives a lower logarithmic aggregation at fixed total response.

A.4 NUMERICAL FORM AND RANKING CONSISTENCY ON FINITE CANDIDATE SETS

The theoretical score requires $E_l > 0$. In practice,

$$S_{\log,\epsilon} = \sum_l \log(E_l + \epsilon), \qquad \epsilon > 0.$$

For any $E_l > 0$,

$$\log(E_l + \epsilon) - \log E_l = \log\left(1 + \frac{\epsilon}{E_l}\right).$$

Since $0 \le \log(1 + x) \le x$ for $x \ge 0$,

$$0 \le \log(E_l + \epsilon) - \log E_l \le \frac{\epsilon}{E_l},$$

and thus

$$0 \le S_{\log,\epsilon} - S_{\log} \le \epsilon \sum_l \frac{1}{E_l}.$$

Therefore $S_{\log,\epsilon} \to S_{\log}$ as $\epsilon \to 0^+$. Also,

$$\frac{\partial S_{\log,\epsilon}}{\partial E_l} = \frac{1}{E_l + \epsilon} > 0.$$

For $E_i > E_j$, $1/(E_j + \epsilon) > 1/(E_i + \epsilon)$, so lower layer responses retain larger marginal effect.

Under the theoretical form, scaling only layer l by $c > 0$ gives

$$E_l \to cE_l,$$

$$S'_{\log} - S_{\log} = \log(cE_l) - \log E_l = \log c.$$

With stabilization, the change is

$$\log \frac{cE_l + \epsilon}{E_l + \epsilon}.$$

Thus equal relative changes yield equal theoretical score changes regardless of layer.

Consider a finite candidate set $\mathcal{A}$ for which every participating layer satisfies $E_l(A) > 0$. For each A,

$$|S_{\log,\epsilon}(A) - S_{\log}(A)| \le \epsilon \sum_l \frac{1}{E_l(A)}.$$

Let

$$B_0 = \max_{A \in \mathcal{A}} \sum_l \frac{1}{E_l(A)}.$$

Then

$$|S_{\log,\epsilon}(A) - S_{\log}(A)| \le \epsilon B_0.$$

If at least one pair has distinct theoretical scores, define

$$\Delta = \min_{\substack{A,B \in \mathcal{A} \\ S_{\log}(A) \ne S_{\log}(B)}} |S_{\log}(A) - S_{\log}(B)| > 0.$$

When

$$\epsilon < \frac{\Delta}{2B_0},$$

for all $A, B \in \mathcal{A}$,

$$\left|\big(S_{\log,\epsilon}(A) - S_{\log,\epsilon}(B)\big) - \big(S_{\log}(A) - S_{\log}(B)\big)\right| \le 2\epsilon B_0 < \Delta.$$

Hence strict pairwise orderings in the theoretical score cannot reverse under the stabilized form; theoretical ties may be broken numerically. If candidates have different layer counts, use the normalized score

$$\frac{1}{L} S_{\log,\epsilon} = \frac{1}{L} \sum_l \log(E_l + \epsilon)$$

to remove the linear scale difference caused by layer count.