Neural Spectral Capacity: Measuring and Designing Architectures from Network Specification Alone

Chenyu Zhu **Ruoyu Zhao** **Zhichao Lu***
Department of Computer Science, City University of Hong Kong

Abstract

Modern Transformer design and compression both reduce to allocating resources under a budget. The standard scalars for these decisions, #Params and #FLOPs, capture size and compute but not architectural structure—two architectures with identical parameter budgets but different depth-width, head, or FFN allocations receive identical scores yet behave differently. We propose Neural Spectral Capacity (`NSC`), a closed-form scalar grounded in the singular-value spectrum of each weight matrix. Under standard random initialization, the Marchenko–Pastur law renders `NSC` computable from the architectural specification alone, with no model instantiation, data, or gradients. Its layer-wise additive structure admits `NSC-DP`, an exact dynamic-programming solver returning the architecture globally maximizing `NSC` under resource constraints in seconds on a CPU—a guarantee that black-box search over existing training-free proxies cannot provide. Empirically, `NSC` outperforms #Params, #FLOPs, and representative training-free proxies in ranking across seven Transformer and CNN families (on FlexiBERT, τ=0.505 on pairs differing in #Params by $<10\%$, where #Params collapses to 0.082); `NSC-DP` discovers a Transformer-XL architecture on WikiText-103 that beats the human-designed baseline in 2 seconds; and prunes LLaMA-7B to the best 5.7B model across eight commonsense reasoning tasks *without any calibration data*, $\sim$5900$\times$ faster than the strongest training-free proxy baseline. Code is available at https://github.com/Optima-CityU/neural-spectral-capacity.

1 Introduction

Designing or compressing a modern Transformer reduces to a single underlying question: under a fixed parameter or compute budget, how should resources be allocated across architectural components? At pretraining time, this manifests as choices over depth-width tradeoffs, FFN ratio, attention-head sharing (GQA, MQA (Ainslie et al., 2023)), and per-layer expert allocation in MoEs (DeepSeek-AI, 2024; Jiang et al., 2024). At post-training time, structured pruning of frontier models (Ma et al., 2023; Munoz et al., 2024; Xia et al., 2024) poses the same question in reverse: under a deployment budget, which components should be retained? These decisions are currently made by intuition, small-scale ablation, or expensive search. The standard scalar tools available to a practitioner are #Params and #FLOPs, which respectively capture *size* and *compute* but say nothing about *structure*—two architectures with the same parameter budget but different depth-width tradeoffs, head allocations, or FFN ratios receive identical scores yet behave differently when trained or deployed. Yet structure is precisely what an architectural choice *is*. Is there a scalar that captures architectural structure, computable from the specification alone?

Why existing training-free proxies fall short for this question. The closest existing candidates come from training-free proxies for neural architecture search (Abdelfattah et al., 2021; Mellor et al., 2021; Wang, 2025): scalar functions that score architectures at initialization. Two limitations make

*Corresponding author: zhichao.lu@cityu.edu.hk.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

them awkward as resource-allocation tools. First, each proxy instantiates a randomly-initialized network and *measures* something on it (activation patterns, gradient norms, attention focus), with the resulting score depending on (architecture, input): real data, synthetic samples, or random Gaussian vectors are all valid choices, and the score changes with the choice. Second, each proxy is a black-box scalar function of the entire network, so optimizing it under resource constraints requires evolutionary or random search, returning the best architecture *visited* rather than the best architecture *possible* under the proxy. Both limitations are amplified at LLM scale, where instantiating each candidate is expensive and the search space is too large to explore by sampling.

Neural Spectral Capacity. We propose *Neural Spectral Capacity* (NSC), a scalar quantity that is a function of the architectural specification alone: deterministic in the spec, computable without ever instantiating a model. Like #Params and #FLOPs, NSC takes the architectural spec as input and returns a number—no random initialization, no input data, no forward or backward pass, no learned hyperparameters. The construction is grounded in the singular-value spectrum of each weight matrix, which we connect to the mutual information of the corresponding linear Gaussian channel (Telatar, 1999). Per-matrix capacity is summed across the network with multi-head attention decomposed per head, yielding an architecture-level score that is layer-wise additive. Under standard random initialization, the Marchenko–Pastur law (Marčenko and Pastur, 1967) renders the expected per-matrix capacity exactly computable from the matrix's dimensions and initialization variance alone, so the entire score reduces to a closed-form expression in architectural parameters.

Two practical advantages of NSC. NSC's structure has two practical consequences that competing proxies do not jointly admit. First, NSC *discriminates* between architectures where #Params and FLOPs cannot. At a fixed parameter or compute budget—the typical setting for resource-allocation decisions in practice—all candidate architectures have nearly identical #Params and FLOPs by construction, so neither quantity can distinguish them. NSC is sensitive to structural choices—depth, head allocation, FFN ratio—that #Params and FLOPs do not see, and assigns different scores accordingly. Second, NSC's layer-wise additivity admits a structural property no other quality-tracking proxy provides: the resource-constrained maximization of NSC reduces to a bounded knapsack solvable exactly by dynamic programming. We exploit this with NSC-DP (Algorithm 1), which returns the architecture *globally maximizing* NSC under resource constraints—both globally optimal under the proxy and orders of magnitude faster than heuristic alternatives, on a single CPU core for spaces of up to 10^{32} candidate architectures.

The contributions of this work are threefold:

- **NSC, a closed-form architectural scalar for Transformers** that is deterministic in the architectural spec—no instantiation, no input data, no gradient computation. Grounded in the singular-value spectrum of each weight matrix, NSC reduces under standard random initialization to a deterministic expression in dimensions and initialization variance alone via the Marchenko–Pastur law (Theorem 1, §3.1–§3.2). Unlike #Params and #FLOPs, NSC captures architectural structure—depth, head allocation, FFN ratio—that size and compute miss.
- **NSC-DP, an exact dynamic-programming solver enabled by NSC's additive structure** (§3.4). NSC-DP returns the architecture *globally maximizing* NSC subject to resource constraints—a guarantee that black-box search (evolutionary, RL, random) over non-decomposable proxies fundamentally cannot provide. Among architectural scalars, only NSC jointly tracks quality and decomposes additively: #Params is additive but does not track quality, and existing training-free proxies track quality but are non-decomposable.
- **Empirical validation across ranking, design, and compression.** Across seven Transformer and CNN families, NSC ranks architectures more accurately than #Params, #FLOPs, and training-free proxies in microseconds on CPU—on FlexiBERT, retaining τ=0.505 on pairs within 10% of #Params, where #Params drops to 0.082 (§4.1). On Transformer-XL, NSC-DP beats the human-designed baseline in 2 seconds, over 400× faster than training-free proxies with heuristic search (§4.2). On LLaMA-7B pruning to 5.7 B, NSC-DP produces the best model across eight commonsense tasks *without calibration data*, ∼5900× faster than the strongest baseline (§4.3).

2 Related work

Random matrix theory in deep learning. Random matrix theory (RMT) has been applied to the spectra neural networks *give rise to*: trained weight matrices (Martin and Mahoney, 2021), activation

Gram matrices (Pennington and Worah, 2017), input–output Jacobians (Pennington et al., 2017), loss Hessians (Pennington and Bahri, 2017), and neural tangent kernels (Fan and Wang, 2020), alongside precise generalization asymptotics for high-dimensional linear models (Advani et al., 2020; Mei and Montanari, 2022); see Couillet and Liao (2022) for a survey. Three related RMT studies analyze networks at initialization, during training, and after training; Table 6 in Appendix A.1 compares these approaches. Martin and Mahoney (2021) read heavy-tailed departures of trained-weight spectra from the Marchenko–Pastur (MP) law as a diagnostic of training quality; at initialization, where NSC operates, that diagnostic is degenerate by construction. Pennington and Worah (2017) derive the limiting activation spectrum at initialization, with the nonlinearity and the input distribution both entering the answer; NSC abstracts from both. Berlyand et al. (2023) prune singular values inside the MP bulk during training, reading them as residual initialization randomness; at initialization the entire spectrum lies in the bulk, so their criterion classifies as noise exactly the mass NSC counts—noise relative to learned structure, budget relative to what a specification provides for training to use. NSC uses the MP law as a *deterministic equivalent* to evaluate capacity from architectural specifications, rather than as a null model, noise criterion, or baseline to extend (§3.2).

Training-free proxies and architecture search. A line of work scores architectures at initialization to bypass training cost in NAS. Early proxies originate from pruning at initialization—SNIP (Lee et al., 2019), SynFlow (Tanaka et al., 2020)—or from activation diversity (NASWOT (Mellor et al., 2021)). More recent proxies include: TF-TAS (Zhou et al., 2022), W-PCA (Wang, 2025), ZeroLM (Chen et al., 2025), AZ-NAS (Lee and Ham, 2024), and softmax-confidence proxies established alongside the FlexiBERT and GPT-2 benchmarks (Serianni and Kalita, 2023). These methods are commonly paired with heuristic search (Real et al., 2019), which returns the best evaluated candidate without a general guarantee of maximizing the proxy over the search space. Despite the "zero-cost" label, evaluating these proxies still requires instantiated networks and measurements of their activations, gradients, or attention scores, rather than architectural specifications alone.

Architecture assessment from specifications. Another line of work assesses candidate architectures from their specifications alone: NN-Mass (Bhardwaj et al., 2021) and DeepMAD (Shen et al., 2023) use connectivity and analytical expressiveness measures, respectively, for CNN design, while MeRino (Zhao et al., 2025) uses log-determinant projection entropies, precomputed tables, and evolutionary search for Transformers. NSC advances spectral evaluation through a deterministic MP approximation computed directly from matrix dimensions and prescribed initialization variance, avoiding weight sampling and SVD-based table construction. Its additive formulation enables NSC-DP to globally maximize the score over the specified discrete search space under additive resource constraints (§3.4). Additional discussion of LLM pruning appears in Appendix A.2.

3 Neural Spectral Capacity (NSC)

Our approach builds on the connection between the singular-value spectrum of a weight matrix and the information capacity of the corresponding linear Gaussian channel (Telatar, 1999). We formalize this per-matrix quantity as *spectral capacity* (§3.1) and show that under standard random initialization it admits a closed-form expression in matrix dimensions and initialization variance alone (§3.2). Aggregating per-matrix capacities—within layers, then across them—yields NSC (§3.3), whose layer-wise additivity admits the exact dynamic-programming solver NSC-DP for resource-constrained architecture search (§3.4).

3.1 Spectral capacity of a weight matrix

We motivate our per-matrix score by treating each weight matrix as a communication channel and asking how much information can flow through it. For a weight matrix $W \in \mathbb{R}^{m\times n}$ acting as a linear Gaussian channel $y = Wx + z$ with $x \sim \mathcal{N}(0, I_n)$ and $z \sim \mathcal{N}(0, I_m)$ independent, the mutual information admits a closed-form expression in terms of W alone (Telatar, 1999):

$$I(x; y) = \tfrac{1}{2} \ln \det(I + W^\top W) = \tfrac{1}{2} \sum_i \ln(1 + \sigma_i^2), \tag{1}$$

where $\{\sigma_i\}$ are the singular values of W (proof in Appendix D.1; we use $\ln$ throughout, measuring information in nats). The decomposition into a sum over singular values follows from the SVD: in the basis of right singular vectors, the channel splits into $\min(m, n)$ *parallel sub-channels*, the

i-th contributing $\frac{1}{2}\ln(1+\sigma_i^2)$. We take the log-determinant itself as the per-matrix score, dropping the constant factor $\frac{1}{2}$: every downstream use of the score—ranking, additive aggregation, exact maximization—is invariant to positive scaling.

Definition 1 (Spectral Capacity)**.**

$$\psi(W) \;:=\; \ln\det\!\big(I + W^\top W\big) \;=\; \sum_{i=1}^{\min(m,n)} \ln(1+\sigma_i^2). \tag{2}$$

Zero singular values contribute zero, so ψ remains well-defined for rank-deficient matrices.

What the channel model assumes—and what it does not. The Gaussian assumptions attach to the probe signal x and the noise z, not to the network's data or weights. The isotropic input $x \sim \mathcal{N}(0, I_n)$ places unit power in every input direction, so each parallel sub-channel above is probed at signal-to-noise ratio σ_i^2: this choice is what makes ψ a function of the singular values alone, and is also the optimal input when no realization of W is available (Telatar, 1999)—our specification-only setting. The channel serves to define ψ; we do not claim that the network's forward pass realizes it.

ψ depends on the full spectrum, not just the parameter count. A natural concern is whether ψ is just a re-expression of model size. It is not. Two matrices with the same dimensions (m, n)—hence the same parameter count mn—can have arbitrarily different spectral capacities depending on their singular-value spectrum. For instance, a full-rank $4096 \times 11\,008$ matrix from a LLaMA-7B FFN layer has $\psi \approx 3174$, while its rank-1 truncation has $\psi \approx 2.5$—a $1\,270\times$ drop with mn unchanged. More generally, ψ is sensitive to architectural changes that parameter count cannot see—spectral redistribution under low-rank adapters, pruning, or head merging, and aspect-ratio differences at fixed mn (Appendix B).

Why does ψ_0 predict trained performance? A second concern is that ψ is computed from randomly initialized weights that training will overwrite. Let ψ_t denote ψ summed over the network's weight matrices after t training steps, with $t = T$ at convergence. Figure 1 shows that across training: (i) ψ_t increases monotonically as spectral structure develops; (ii) rankings at $t = 0$ are preserved throughout ($\rho(\psi_0, \psi_T) = 1.00$); (iii) converged ψ_T ranks final PPL perfectly ($|\rho| = 1.00$). The perfect correlations reflect the small sample of 10 architectures with deliberate size spread; larger-scale results in §4 report Spearman $\rho \in [0.78, 0.97]$. A partial mechanistic explanation: $\partial\psi/\partial\sigma_i = 2\sigma_i/(1+\sigma_i^2)$ peaks at $\sigma_i = 1$ (Appendix D.2), where variance-preserving initialization places the bulk of the spectrum (Glorot and Bengio, 2010; He et al., 2015).

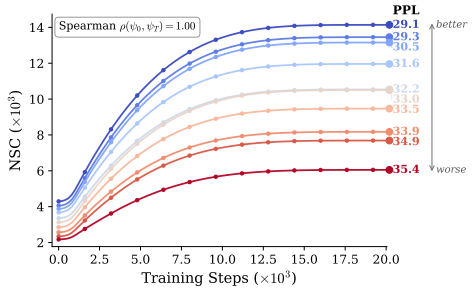


Figure 1: Spectral capacity tracks learning. Curves show ψ_t across training for 10 GPT-2 architectures on WikiText-103; color and right-axis label indicate final perplexity. Initialization-time ranking is preserved throughout training, and converged values correlate with PPL.

3.2 Spectral capacity in closed form

Definition 1 requires the singular-value spectrum of W, which appears to demand instantiating the matrix and computing an SVD. We now show that under standard random initialization, ψ admits a deterministic expression depending only on the matrix's dimensions and initialization variance.

The Marchenko–Pastur form of ψ. By the Marchenko–Pastur theorem (Marčenko and Pastur, 1967), when $W \in \mathbb{R}^{m\times n}$ has i.i.d. entries with mean zero and variance s^2, the empirical distribution of the normalized squared singular values $\sigma_i^2/(Ms^2)$ converges, as $\min(m,n) \to \infty$, to a deterministic law with density $f_{\mathrm{MP}}(\lambda;\gamma)$, where $M = \max(m,n)$, $N = \min(m,n)$, $\gamma = N/M$. This convergence is universal: bounded fourth moment of the entries suffices, covering all standard initialization schemes (Xavier, Kaiming, truncated normal) (Bai and Silverstein, 2010). Integrating $\ln(1 + Ms^2\lambda)$ against this law yields the limiting capacity per sub-channel (proof in Appendix D.3):

Theorem 1 (Marchenko–Pastur Spectral Capacity). *Let $W \in \mathbb{R}^{m\times n}$ have i.i.d. entries of mean zero and variance s^2, and let $\min(m,n) \to \infty$ with γ and Ms^2 held fixed. Then*

$$\frac{\psi(W)}{N} \xrightarrow{\text{a.s.}} \mathcal{I}(\gamma, Ms^2) \;:=\; \int_{\lambda_-}^{\lambda_+} \ln\bigl(1 + Ms^2\lambda\bigr)\, f_{\mathrm{MP}}(\lambda;\gamma)\, d\lambda, \tag{3}$$

where $\lambda_\pm = (1 \pm \sqrt{\gamma})^2$.

$\mathcal{I}(\gamma, Ms^2)$ is the *Shannon transform* of the Marchenko–Pastur law at signal-to-noise ratio Ms^2, a standard random-matrix quantity with an explicit closed form (Tulino and Verdú, 2004). We henceforth define

$$\psi(W) \;:=\; \psi_{\mathrm{MP}}(m,n,s) \;:=\; N\,\mathcal{I}\bigl(\gamma,\, Ms^2\bigr). \tag{4}$$

The limit functional of Theorem 1, evaluated at the matrix's own aspect ratio and signal-to-noise parameter, is the definition rather than a finite-sample approximation of an SVD-based quantity. ψ thereby becomes a deterministic function of (m,n,s)—of the matrix's place in the architecture specification—with no dependence on a particular weight realization. At practical dimensions the limit is tight: the relative error of ψ_{MP} against the SVD-based value ψ_W of Definition 1 decays as $O(1/\min(m,n))$, falling below 0.4% at $\min(m,n) = 128$—the smallest hidden dimension across our benchmarks—and to $\sim 10^{-4}$ by $\min(m,n) = 1024$ (Appendix G.1). On the LoNAS-LLaMA-7B trained supernet, ψ_{MP} matches ψ_W at $\tau{=}\rho{=}1.0000$ across all 129 Pareto-optimal subnets, with 0.039% mean SVD-oracle regret and a $720\times$ compute saving over the SVD route (Appendix G.2).

Robustness to initialization choice. Comparing architectures via Eq. 4 requires a single initialization protocol fixed across all candidates, so that score differences reflect architecture rather than initialization. Under a fixed scheme, s is itself determined by the matrix's shape and role—Xavier sets $s^2 = 2/(m+n)$, Kaiming fan-in $s^2 = 2/n$, truncated normal a constant—so ψ_{MP} effectively reduces to a function of the dimensions alone. The choice of fixed scheme broadly preserves the resulting ranking (Appendix H).

3.3 From single matrices to layers and networks

The spectral capacity ψ scores one weight matrix. To score an architecture, we fix which matrices are considered, then aggregate—first within a layer, then across layers.

Which matrices are considered. We consider the network's linear-projection weight matrices: per-head attention projections, the attention output projection, and FFN matrices. For multi-head attention with H heads and width d, we consider each head's Q/K/V projections separately rather than treating $W_{QKV} \in \mathbb{R}^{3d\times d}$ as one matrix; this mirrors the forward pass (each head computes attention in its own $d_h = d/H$ subspace) and is necessary to discriminate architectures differing in head count. Convolutions are reshaped from $W \in \mathbb{R}^{c_{\text{out}}\times c_{\text{in}}\times k\times k}$ to $\widetilde{W} \in \mathbb{R}^{c_{\text{out}}\times c_{\text{in}}k^2}$—the form in which the convolution acts as a linear map on im2col patches—before applying ψ. Embeddings (dictated by vocabulary and sequence length), biases, normalization, and nonlinearities are excluded.

From matrices to a layer. Within a layer, we aggregate by summation: the capacity of layer l is $\Psi_l := \sum_{j=1}^{J_l} \psi(W_{l,j})$. The sum is itself a spectral capacity: because block-diagonal composition adds log-determinants, $\Psi_l = \psi\bigl(\mathrm{diag}(W_{l,1},\ldots,W_{l,J_l})\bigr)$—the capacity of the layer's matrices composed *in parallel*. Summation is thus the parallel half of the composition rule for channel capacities (Foggo and Yu, 2023), extending the parallel sub-channels of §3.1 from directions within one matrix to matrices within one layer. For the head dimension this parallelism is literal: the heads operate side by side in disjoint subspaces.

Table 1: **Across-layer aggregation ablation.** Spearman ρ against trained performance (GLUE, perplexity, ImageNet Top-1); number of distinct scores in parentheses (500/200/1,001 architectures).

Aggregation	FlexiBERT	GPT-2	AutoFormer-T
$\sum_l \Psi_l$	0.884 (297)	0.968 (200)	0.810 (208)
$\min_l \Psi_l$	0.178 (22)	0.813 (195)	-0.173 (3)
$\prod_l \Psi_l$	0.917 (362)	0.712 (200)	0.816 (458)

From layers to a network. Summing layer capacities gives the network-level score:

Algorithm 1 NSC-DP: NSC-guided Dynamic Programming

Input: Network-level set $\mathcal{G}$; layer-level sets $\{\mathcal{X}_l(g)\}$; resource budget B
Output: Architecture a^* maximizing NSC under budget B

best $\leftarrow -\infty$; $a^* \leftarrow$ NONE
for each network-level configuration $g \in \mathcal{G}$ **do**
  $L \leftarrow L(g)$ {number of layers determined by g}
  Precompute per-layer capacities $\Psi_l(\cdot\,; g)$ and costs $c_l(\cdot\,; g)$ {cached ψ_{MP}}
  Solve by dynamic programming, returning maximizer $\{x_l^*(g)\}_{l=1}^{L}$ and value

$$\hat{\Psi}(g) \leftarrow \max_{\substack{x_l \in \mathcal{X}_l(g) \\ l=1,\dots,L}} \sum_{l=1}^{L} \Psi_l(x_l; g) \quad \text{subject to} \quad \sum_{l=1}^{L} c_l(x_l; g) \le B.$$

  if $\hat{\Psi}(g) >$ best **then**
    $a^* \leftarrow (g,\, x_1^*(g), \dots, x_L^*(g))$; best $\leftarrow \hat{\Psi}(g)$
  end if
end for
return a^*

Definition 2 (Neural Spectral Capacity)**.** *For a network f with L layers:*

$$\mathit{NSC}(f) \;=\; \sum_{l=1}^{L} \Psi_l \;=\; \sum_{l=1}^{L} \sum_{j=1}^{J_l} \psi_{\mathrm{MP}}(m_{l,j},\, n_{l,j},\, s_{l,j}). \tag{5}$$

The second equality uses the closed form of §3.2, making NSC evaluable in $O(L)$ from the specification alone. Across layers, parallelism has no literal counterpart: layers execute in sequence, as do the attention and FFN blocks within each layer. Summing across them is therefore an empirically motivated modelling choice, not a claim about the end-to-end mutual information of the forward pass. Table 1 supports this choice: summation ranks best on average and no alternative beats it on more than one benchmark, while the bottleneck (min) rule collapses—assigning the 500 FlexiBERT architectures only 22 distinct scores and turning anti-correlated on AutoFormer-Tiny; Appendix C gives the full protocol.

3.4 Architecture design via NSC-DP

Coupling NSC with structured optimization—enabled by its layer-wise additive form—yields **NSC-guided Dynamic Programming (NSC-DP, Algorithm 1)**, a solver that returns the architecture globally maximizing NSC under a resource budget. The construction exploits a two-level decomposition. A Transformer architecture decomposes as $a = (g, x_1, \dots, x_{L(g)})$, where $g \in \mathcal{G}$ denotes *network-level* decisions (depth, stage layout, embedding size, etc.) and $x_l \in \mathcal{X}_l(g)$ denotes *layer-level* decisions at layer l (heads, FFN width, MLP ratio, etc.). The network-level g both fixes the number of layers $L(g)$ and enters the layer capacities (e.g., via embedding size). Conditioned on g, NSC separates across layers: $\mathrm{NSC}(a) = \sum_{l=1}^{L(g)} \Psi_l(x_l; g)$, each layer's capacity depending only on its own decisions x_l. Standard deployment constraints admit the same per-layer structure: $R(a) = \sum_{l=1}^{L(g)} c_l(x_l; g) \le B$, where R can represent parameter count, FLOPs, or any resource that decomposes as a sum of per-layer costs. Once g is fixed, the layer-level problem is therefore a knapsack—bounded or multiple-choice, depending on the search space—solvable exactly by dynamic programming. Algorithm 1 enumerates $g \in \mathcal{G}$ and solves this inner knapsack for each g. This exact-optimization structure gives NSC-DP four properties: **Global optimality.** Because the outer loop exhausts $\mathcal{G}$ and the inner DP solves the layer-level knapsack exactly, NSC-DP returns the architecture that *globally maximizes NSC* subject to the resource budget—global with respect to the proxy objective, not the underlying architecture design problem. By contrast, existing training-free proxies serve as black-box fitness functions and return only the best architecture visited by heuristic search.

Multi-budget search. The DP table additionally encodes the proxy-optimum at every reachable budget $b \le B$, so a single NSC-DP run yields the entire Pareto front over the discretized budget grid as

$\mathcal{O}(B)$ table look-ups—a property absent from training-free proxy baselines, which require a separate single-objective search per budget (Appendix E).

Efficiency. NSC is closed-form and admits caching: each ψ_{MP} evaluation is a 1D numerical quadrature costing $\sim$10 μs, reused across network-level configurations whenever the (m, n, s) tuple repeats. The inner DP runs in $\mathcal{O}(L(g) \cdot B \cdot |\mathcal{X}_l(g)|)$ on a discretized budget axis—negligible compared to a single forward pass. Combined, NSC-DP searches spaces of up to 10^{32} architectures in seconds on a single CPU core, with no GPU, no data, and no model instantiation.

Generality. NSC-DP applies to any search space where (i) the NSC score decomposes as a sum of independent per-layer contributions, and (ii) the resource constraint is a sum of per-layer costs. These conditions hold broadly: in AutoFormer, the layer-level variables are MLP ratio and head count; in LoNAS-LLaMA, FFN width and LoRA rank.

4 Experiments

We evaluate NSC in three settings: (i) ranking quality across seven Transformer and CNN architecture families (§4.1), (ii) end-to-end neural architecture search on Transformer-XL and AutoFormer (§4.2), and (iii) structured pruning of LLaMA-7B (§4.3). Table 2 summarizes the architecture families used across the three settings.

Table 2: Architecture families and decision dimensions evaluated in our experiments. *Decision Dimensions* lists the architectural axes that vary across candidate networks within each family; *#Archs.* denotes the (estimated) total number of distinct architectures.

	Architecture	Decision Dimensions	#Archs.
Transformers	BERT (Serianni and Kalita, 2023)	Hidden dim, depth, per-layer FFN dim & attn. type	1.1×10^7
	GPT-2 (Javaheripi et al., 2022)	Embed dim, depth, per-layer FFN dim & heads	$> 10^{54}$
	Transformer-XL (Javaheripi et al., 2022)	Embed dim, depth & per-layer FFN dim	2×10^{25}
	ViT (AutoFormer) (Chen et al., 2021a)	Embed dim, depth, per-layer heads & MLP ratio	1.7×10^{16}
	LLaMA (LoNAS) (Munoz et al., 2024)	Per-layer FFN dim & LoRA rank (32 layers)	1.0×10^{32}
CNNs	CNN (NATS-Bench) (Dong et al., 2022)	Per-layer channels (5 layers)	32,768
	MobileNetV3 (OFA) (Cai et al., 2020)	Per-stage depth, per-layer channels & kernel size	2.0×10^{19}

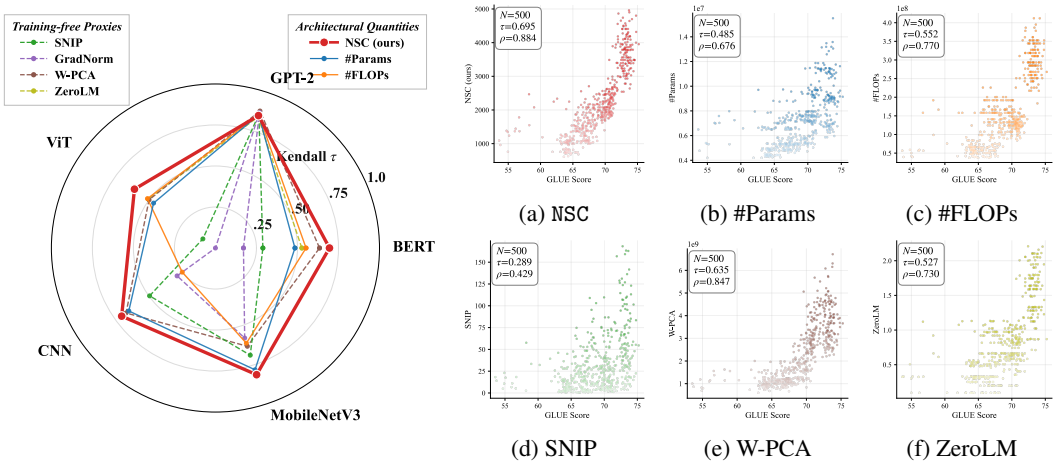
4.1 Ranking quality across architecture families via NSC

Before applying NSC to architecture search (§4.2) and LLM pruning (§4.3), we first establish that it tracks trained performance across architecture families. This subsection asks: does NSC, computed from the architectural specification alone, rank candidate architectures more accurately than #Params, #FLOPs, and representative training-free proxies that require model instantiation, forward or backward passes, and sampled inputs?

Setup. We evaluate NSC on three Transformer architecture families: 500 BERT architectures from FlexiBERT (Serianni and Kalita, 2023; Tuli et al., 2023), 200 GPT-2 architectures from LiteTransformerSearch (Javaheripi et al., 2022), and 1,001 ViT architectures from the AutoFormer-Tiny supernet (Chen et al., 2021a). For FlexiBERT, we use the full decision space (dynamic hidden size and FFN), differing from the constrained setting in Serianni and Kalita (2023) (fixed hidden size and FFN); results under the original setting are reported in Appendix F. To assess whether NSC generalizes beyond Transformers, we additionally include two CNN architecture families: CNN architectures from NATS-Bench-SSS (Dong et al., 2022) and MobileNetV3 architectures from the OFA supernet (Cai et al., 2020). We compare NSC against the two existing architectural

Table 3: **Ranking correlation on FlexiBERT.** *10%-PW*: τ on architecture pairs with $<10\%$ difference in #Params. *Time*: total wall-clock for 500 architectures. **Bold**/underline: best/second best.

Method	τ	10%-PW τ	Time
NSC (ours)	**0.695**	**0.505**	**2 ms**
#Params	0.485	0.082	—
#FLOPs	0.552	0.329	2 ms
W-PCA	<u>0.635</u>	<u>0.417</u>	88 s
ZeroLM	0.527	0.355	63 s
SNIP	0.289	0.237	73 s
GradNorm	0.171	0.164	74 s



(a) NSC (b) #Params (c) #FLOPs

(d) SNIP (e) W-PCA (f) ZeroLM

Figure 2: Ranking quality of NSC against architectural quantities (#Params, #FLOPs) and training-free proxies across five architecture families. *Left:* Kendall τ on FlexiBERT, GPT-2, AutoFormer-Tiny, NATS-Bench-SSS, and MobileNetV3. *Right:* Scatter of six scoring functions against GLUE score on 500 BERT architectures from FlexiBERT, with Kendall τ and Spearman ρ annotated.

quantities (#Params, #FLOPs) and four representative training-free proxies (W-PCA (Wang, 2025), ZeroLM (Chen et al., 2025), SNIP (Lee et al., 2019), GradNorm (Abdelfattah et al., 2021)); benchmark-specific additional baselines are evaluated in Appendix F. Wall-clock times are measured on a single A100 GPU for training-free proxies and a single CPU core for architectural quantities including NSC.

NSC captures architectural quality beyond size and compute. On FlexiBERT, NSC reaches $\tau = 0.695$ versus 0.485 for #Params and 0.552 for #FLOPs (Table 3); under the 10%-PW control—which restricts τ to architecture pairs differing in #Params by less than 10%, removing size as a discriminating signal—#Params drops to $\tau = 0.082$ while NSC retains $\tau = 0.505$, suggesting that NSC captures architectural information beyond parameter count. This pattern generalizes: NSC's 10%-PW τ exceeds #Params' on every benchmark—e.g., 0.503 vs. 0.322 on AutoFormer-Tiny, 0.372 vs. 0.116 on NATS-Bench-SSS, 0.535 vs. 0.414 on MobileNetV3 (Appendix F.2). NSC additionally outperforms training-free proxies that require model instantiation, forward or backward passes on sampled inputs, or per-dataset proxy search, despite being computed from the architecture specification alone (Figure 2; full per-family results in Appendix F). On FlexiBERT it leads W-PCA by $+0.060$ and ZeroLM by $+0.168$ in τ; the lead widens on AutoFormer-Tiny ($+0.110$ over W-PCA), while on GPT-2 NSC is competitive with the leading proxies (within 0.03 of W-PCA) at orders-of-magnitude lower wall-clock. NSC also top-ranks both CNN architecture families (NATS-Bench-SSS, MobileNetV3), despite being designed for Transformers. All of these results are obtained at microsecond cost per architecture (Table 3, *Time* column), four orders of magnitude faster than the closest training-free competitors.

4.2 Architecture search via NSC-DP

The ranking results above establish that NSC tracks trained performance broadly. We now ask whether *optimizing* NSC produces good architectures—first in the classical NAS setting (this section), then via structured pruning of a pretrained LLM (§4.3).

Setup. We instantiate NSC-DP on two Transformer families: Transformer-XL (TXL) (Dai et al., 2019) from LiteTransformerSearch (Javaheripi et al., 2022), and vision Transformers from AutoFormer (Chen et al., 2021a) (Table 2). For each benchmark, all baselines and NSC-DP are evaluated under an identical protocol: for TXL, every selected architecture is trained from scratch on WikiText-103 under the same hyperparameters; for AutoFormer-Tiny, every selected architecture inherits weights from the AutoFormer supernet. The TXL search uses a non-embedding parameter budget of $38.4\,\text{M} \pm 5\%$ matching TXL Base; AutoFormer-Tiny uses a parameter budget of $5.7\,\text{M}$.

NSC-DP instantiation and baselines. For Transformer-XL, we instantiate Algorithm 1 with network-level decisions $g = (d_{\text{model}}, L)$ and layer-level decisions $x_l = d_{\text{ff},l}$; the inner problem is a bounded

knapsack with additive value $\Psi_l = 2\,\psi_{\mathrm{MP}}(d_{\mathrm{ff},l}, d_{\mathrm{model}}, s)$ (the FFN pair; attention terms are fixed by g) and additive cost $c_l = d_{\mathrm{ff},l}(2d_{\mathrm{model}}{+}1)$, with the resource budget B matching the parameter range above. For AutoFormer-Tiny, the network-level decisions are network depth and embedding dimension, with layer-level decisions $x_l = (\text{MLP ratio}_l, \text{heads}_l)$; the inner problem is a multiple-choice knapsack solved analogously by dynamic programming. Baselines reflect each benchmark's prior literature: for Transformer-XL, we compare against the original Transformer-XL base and three training-free proxies—Synaptic Diversity (Zhou et al., 2022), Softmax Confidence (Serianni and Kalita, 2023), and W-PCA (Wang, 2025)—each used as fitness for evolutionary search; for AutoFormer-Tiny we compare against the AutoFormer-Tiny oracle (Chen et al., 2021a), TF-TAS (Zhou et al., 2022), AZ-NAS (Lee and Ham, 2024), and W-PCA proxy used as fitness for evolutionary search. All compared baselines run on GPU while `NSC-DP` runs on a single CPU core.

Results. Table 4 reports both experiments. On Transformer-XL, `NSC-DP` discovers an architecture with test PPL 23.087, the lowest among all methods—improving over Transformer-XL Base (23.279) and Synaptic Div. (23.135). The search completes in 2.0 seconds on CPU, more than $400\times$ faster than the 883–965 s consumed by training-free proxy baselines. On AutoFormer-Tiny, `NSC-DP` matches the day-scale AutoFormer oracle within 0.03 accuracy points while reducing search cost by six orders of magnitude, and outperforms all heuristics-based alternatives on Acc@1. Results on AutoFormer-Small/Base are reported in Appendix F.3 and follow the same pattern; Appendix F.5 analyzes structural properties of `NSC-DP`'s discovered architectures, including a proof that uniform per-layer FFN allocation is the proxy-optimum on Transformer-XL.

Table 4: End-to-end architecture search by `NSC-DP` on language (*left*: Transformer-XL on WikiText-103) and vision (*right*: AutoFormer-Tiny on ImageNet-1K) Transformers. #P: non-embedding parameters in millions. Search-cost: s = seconds, d = GPU-days. **Bold**: best; underline: second best.

Method	#P (M)	Test PPL (↓)	Cost
TXL Base	38	23.279	—
Synaptic Div.	37	<u>23.135</u>	903 s
Softmax Conf.	39	23.532	883 s
W-PCA	40	23.669	965 s
`NSC-DP` **(ours)**	40	**23.087**	**2.0 s**

Method	#P (M)	Acc@1 (↑)	Acc@5 (↑)	Cost
AutoFormer-T	5.7	**75.308**	92.690	24 d
TF-TAS	5.9	75.234	<u>92.730</u>	0.5 d
AZ-NAS	5.9	74.804	92.504	2,592 s
W-PCA	5.8	74.752	92.500	206 s
`NSC-DP` **(ours)**	5.8	<u>75.276</u>	**92.788**	**0.03 s**

4.3 Structured pruning of LLaMA-7B via `NSC-DP`

We now apply `NSC-DP` to LLM compression: structured pruning of a pretrained LLM under a deployment budget. This setting is qualitatively distinct from the end-to-end NAS of §4.2: rather than designing a network from scratch, we compress a pretrained LLM by selecting which components to retain. The training-free, calibration-data-free property of `NSC` is materially differentiating here—existing structured-pruning methods couple the pruning decision to the pretrained weights and a calibration mini-batch, while `NSC` scores pruning configurations from the specification of the pruned subnetwork alone.

Setup. We evaluate `NSC-DP` on structured pruning of LLaMA-7B via the LoNAS SuperNet of Munoz et al. (2024), with parameter budget 5.7 B. The SuperNet exposes two elastic axes per Transformer block: per-block LoRA rank $r_\ell \in \{32, 28\}$ applied to attention and FFN projections, and per-block FFN intermediate dimension $h_\ell \in \{11008, 9632, 8256, 6880, 5504\}$.

`NSC-DP` **instantiation and baselines.** The pruning decision space contains only layer-level decisions: each block selects a tuple $x_\ell = (r_\ell, h_\ell)$ of LoRA rank and FFN intermediate dimension. With network-level depth and embedding dimension fixed by the LLaMA-7B architecture, `NSC-DP` reduces to a single multiple-choice knapsack solved by exact dynamic programming over the closed-form ψ_{MP} cache. We compare against four training-free proxies—SNIP (Lee et al., 2019), GradNorm (Abdelfattah et al., 2021), SynFlow (Tanaka et al., 2020), and W-PCA (Wang, 2025)—each used as the fitness function for an evolutionary search under the same 5.7 B budget (population size 50 for 20 generations, yielding 1,000 proxy evaluations total per method). Each proxy is computed on a calibration mini-batch from WikiText-103 ($B{=}2$ sequences of length $T{=}2048$); `NSC-DP` requires no calibration data. The LoNAS-SuperNet (the un-pruned 6.7 B configuration) serves as the upper-bound reference. All baselines run on a GPU; `NSC-DP` runs on a single CPU thread. We evaluate pruned networks under a unified zero-shot multiple-choice protocol following Munoz et al. (2024)

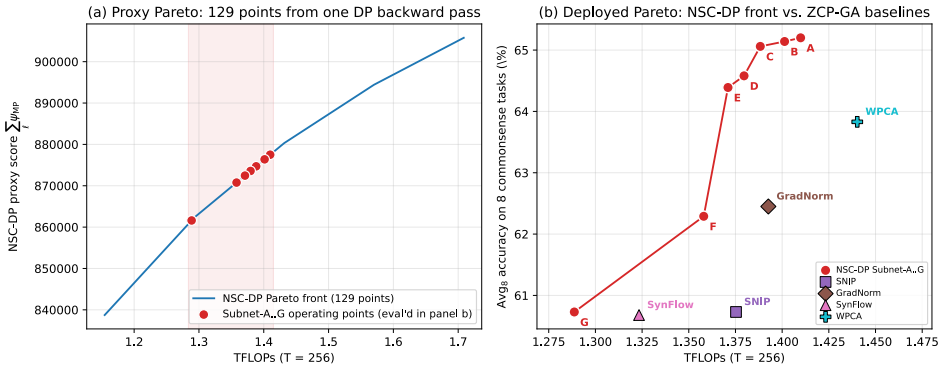


Figure 3: **Pareto front from a single `NSC-DP` run.** Avg$_8$ commonsense reasoning vs. TFLOPs (at $T{=}256$) for the seven LoNAS-LLaMA-7B operating points (A–G), all read from the same 0.46 s DP backward pass. Full numerical results in Appendix E.

Table 5: Pruning LLaMA-7B from 6.7 B to a 5.7 B target via the LoNAS SuperNet: `NSC-DP` versus training-free proxy baselines. The LoNAS-SuperNet row reports the un-pruned performance for reference. *Cost*: search-phase wall-clock (s = seconds, min = minutes). **Bold**: best per column.

Method	**Params**	BoolQ	PIQA	SIQA	HSwag	WGrnd	ARC-e	ARC-c	OBQA	**Avg$_8$**	**Cost**
LoNAS-SuperNet	6.7 B	66.02	77.58	72.93	57.49	66.85	78.41	62.29	76.20	69.72	—
SNIP	5.4 B	61.93	69.21	64.99	41.26	62.04	66.50	52.47	67.40	60.73	22 min
SynFlow	5.2 B	61.13	70.24	63.25	45.94	59.98	66.84	51.88	66.20	60.68	23 min
GradNorm	5.5 B	61.31	70.78	67.35	46.33	62.59	69.02	55.03	67.20	62.45	22 min
W-PCA	5.7 B	63.82	70.95	67.91	49.24	62.75	69.57	56.40	70.00	63.83	45 min
`NSC-DP` **(ours)**	**5.7 B**	**64.01**	**72.85**	**68.94**	**51.30**	**64.72**	**72.43**	**58.70**	**70.80**	**65.47**	**0.5 s**

on eight commonsense reasoning tasks (BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-Easy, ARC-Challenge, OpenBookQA) and report the unweighted average Avg$_8$.

Results. Table 5 reports four findings. **(1) Best pruned model at matched parameter budget.** `NSC-DP` produces the highest-quality pruned model at the 5.7 B budget, outperforming W-PCA by $+1.64$ pp on Avg$_8$ and dominating every baseline on every individual task. **(2) Specification-based selection at LLM scale.** The four baselines rely on the LoNAS-LLaMA-7B weights and a calibration mini-batch; `NSC-DP` selects from the architectural specification alone, requiring neither. **(3) Search efficiency.** `NSC-DP` completes pruning in 0.46 s on a CPU thread, $\sim$2,800$\times$ faster than the fastest baseline (GradNorm, 22 min) and $\sim$5,900$\times$ faster than the strongest baseline (W-PCA, 45 min). **(4) Pareto-front output at zero marginal cost.** The same DP backward pass that produces the 5.7 B result encodes the optimum at every reachable budget; reading off the seven LoNAS-LLaMA operating points spanning 1.29–1.41 TFLOPs requires no additional search (Figure 3; Appendix E).

5 Conclusion

We introduced Neural Spectral Capacity (`NSC`), a closed-form architectural scalar computable from a Transformer's specification alone, and `NSC-DP`, an exact dynamic-programming solver returning the architecture globally maximizing `NSC` under resource constraints. Empirically, `NSC` ranks architectures more accurately than #Params, #FLOPs, and representative training-free proxies across seven Transformer and CNN families; `NSC-DP` discovers a Transformer-XL architecture beating the human-designed baseline in 2 seconds on a single CPU core, and produces the best pruned LLaMA-7B at 5.7 B across eight commonsense reasoning tasks *without any calibration data*, $\sim$5900$\times$ faster than the strongest baseline. Two further architectural axes extend by construction: per-head decomposition (§3.3) scores MHA/GQA/MQA grouping, and the layer-wise additive sum scores per-layer expert allocation in MoEs. We see `NSC` growing into a general quantitative tool for resource-constrained Transformer architecture choice.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and constructive feedback. The authors declare no competing interests.

References

Mohamed S Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas D Lane. Zero-cost proxies for lightweight NAS. In *Int. Conf. Learn. Represent.*, 2021.

Madhu S. Advani, Andrew M. Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Netw.*, 132:428–446, 2020.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2023.

Zhidong Bai and Jack W. Silverstein. *Spectral Analysis of Large Dimensional Random Matrices*. Springer, 2 edition, 2010.

Leonid Berlyand, Etienne Sandier, Yitzchak Shmalo, and Lei Zhang. Enhancing accuracy in deep learning using random matrix theory. *arXiv preprint arXiv:2310.03165*, 2023.

Kartikeya Bhardwaj, Guihong Li, and Radu Marculescu. How does topology influence gradient propagation and model performance of deep networks with DenseNet-type skip connections? In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 13498–13507, 2021.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *Int. Conf. Learn. Represent.*, 2020.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. AutoFormer: Searching Transformers for visual recognition. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021a.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on ImageNet in four GPU hours: A theoretically inspired perspective. In *Int. Conf. Learn. Represent.*, 2021b.

Zhen-Song Chen, Hong-Wei Ding, Xian-Jia Wang, and Witold Pedrycz. ZeroLM: Data-free transformer architecture search for language models. *arXiv preprint arXiv:2503.18646*, 2025.

W. J. Conover and Ronald L. Iman. Rank transformations as a bridge between parametric and nonparametric statistics. *Am. Stat.*, 35(3):124–129, 1981.

Romain Couillet and Zhenyu Liao. *Random Matrix Methods for Machine Learning*. Cambridge University Press, 2022.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pages 2978–2988, 2019.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Xuanyi Dong and Yi Yang. Network pruning via transformable architecture search. In *Adv. Neural Inf. Process. Syst.*, 2019.

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. NATS-Bench: Benchmarking NAS algorithms for architecture topology and size. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3634–3646, 2022.

Zhou Fan and Zhichao Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. In *Adv. Neural Inf. Process. Syst.*, 2020.

Brandon Foggo and Nanpeng Yu. On the maximum mutual information capacity of neural architectures. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*, 2023.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Int. Conf. Mach. Learn.*, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int. Conf. Artif. Intell. Stat.*, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1026–1034, 2015.

Mojan Javaheripi, Gustavo de Rosa, Subhabrata Mukherjee, Shital Shah, Tomasz Religa, Caio Cesar Teodoro Mendes, Sebastien Bubeck, Farinaz Koushanfar, and Debadeepta Dey. Litetransformersearch: Training-free neural architecture search for efficient language models. In *Adv. Neural Inf. Process. Syst.*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

M. G. Kendall. Partial rank correlation. *Biometrika*, 32(3/4):277–283, 1942.

Junghyup Lee and Bumsub Ham. AZ-NAS: Assembling zero-cost proxies for network architecture search. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 5893–5903, 2024.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *Int. Conf. Learn. Represent.*, 2019.

Guihong Li, Yuedong Yang, Kartikeya Bhardwaj, and Radu Marculescu. ZiCo: Zero-shot NAS via inverse coefficient of variation on gradients. In *Int. Conf. Learn. Represent.*, 2023a.

Yuhong Li, Jiajie Li, Cong Hao, Pan Li, Jinjun Xiong, and Deming Chen. Extensible and efficient proxy for neural architecture search. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pages 6199–6210, 2023b.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *Adv. Neural Inf. Process. Syst.*, 2023.

V.A. Marčenko and Leonid Pastur. Distribution of eigenvalues for some sets of random matrices. *Math. USSR Sb.*, 1:457–483, 1967.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *J. Mach. Learn. Res.*, 22(165):1–73, 2021.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Commun. Pure Appl. Math.*, 75(4):667–766, 2022.

Joe Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. In *Int. Conf. Mach. Learn.*, 2021.

Juan Pablo Munoz, Jinjie Yuan, Yi Zheng, and Nilesh Jain. LoNAS: Elastic low-rank adapters for efficient large language models. In *Proc. Joint Int. Conf. Comput. Linguist. Lang. Resour. Eval.*, pages 10760–10776, 2024.

Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Int. Conf. Mach. Learn.*, 2017.

Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. In *Adv. Neural Inf. Process. Syst.*, 2017.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In *Adv. Neural Inf. Process. Syst.*, 2017.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proc. AAAI Conf. Artif. Intell.*, volume 33, pages 4780–4789, 2019.

Aaron Serianni and Jugal Kalita. Training-free neural architecture search for RNNs and transformers. In *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2023.

Xuan Shen, Yaohua Wang, Ming Lin, Yilun Huang, Hao Tang, Xiuyu Sun, and Yanzhi Wang. DeepMAD: Mathematical architecture design for deep convolutional neural network. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pages 6163–6173, 2023.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *Int. Conf. Learn. Represent.*, 2024.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Adv. Neural Inf. Process. Syst.*, 2020.

Emre Telatar. Capacity of multi-antenna gaussian channels. *Eur. Trans. Telecommun.*, 10(6):585–595, 1999. doi: 10.1002/ett.4460100604.

Shikhar Tuli, Bhishma Dedhia, Shreshth Tuli, and Niraj K. Jha. FlexiBERT: Are current transformer architectures too homogeneous and rigid? *J. Artif. Intell. Res.*, 77:39–70, 2023. doi: 10.1613/jair.1.13942.

Antonia M. Tulino and Sergio Verdú. *Random Matrix Theory and Wireless Communications*, volume 1 of *Foundations and Trends in Communications and Information Theory*. Now Publishers, Hanover, MA, 2004.

Shang Wang. W-PCA based gradient-free proxy for efficient search of lightweight language models. In *Int. Conf. Learn. Represent.*, 2025.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *Int. Conf. Learn. Represent.*, 2024.

Tomomasa Yamasaki, Zhehui Wang, Tao Luo, Niangjun Chen, and Bo Wang. RBFleX-NAS: Training-free neural architecture search using radial basis function kernel and hyperparameter detection. *IEEE Trans. Neural Netw. Learn. Syst.*, 36(6):10057–10071, 2025.

Youpeng Zhao, Ming Lin, Huadong Tang, Qiang Wu, and Jun Wang. MeRino: Entropy-driven design for generative language models on IoT devices. In *Proc. AAAI Conf. Artif. Intell.*, pages 22840–22848, 2025.

Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Xing Sun, Yonghong Tian, Jie Chen, and Rongrong Ji. Training-free transformer architecture search. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022.

A Additional related work

A.1 Comparison of RMT-based approaches in deep learning

Table 6 compares the approaches discussed in §2 by their stage of application, required information, use of the Marchenko–Pastur law, and outputs.

Table 6: **Random matrix theory in deep learning: what each method needs and produces.** Rows are ordered by stage of application, from architectural specification to analysis of trained networks.

	When it applies	Needs beyond the architecture	How it uses the MP law	What it produces
`NSC` **(ours)**	before any network exists	nothing—matrix shapes and initialization variances suffice	as the quantity itself: the MP spectrum is the capacity being counted	closed-form score + exact argmax (`NSC-DP`)
Pennington and Worah (2017)	at initialization	nonlinearity + input distribution	the linear baseline their nonlinear theory extends	limiting spectral density of activations
Berlyand et al. (2023)	during training	the partially trained weights	noise criterion: MP-bulk singular values are pruned	a pruned network
Martin and Mahoney (2021)	after training	the trained checkpoint	the null model that training departs from	a training-quality diagnostic

A.2 Structured pruning of large language models

A growing body of work compresses pretrained LLMs along structured axes under a deployment budget. LLM-Pruner (Ma et al., 2023) prunes coupled structures via gradient-based importance on a small calibration set; Sheared LLaMA (Xia et al., 2024) jointly learns pruning masks and continues pretraining; LoNAS (Munoz et al., 2024) expresses pruning as supernet-based architecture selection. Magnitude- and saliency-based one-shot pruners (SparseGPT (Frantar and Alistarh, 2023), Wanda (Sun et al., 2024)) require calibration activations to score individual weights. `NSC` positions in this landscape as a score function that operates from the architectural specification of the pruned subnetwork alone, requiring neither the pretrained weights nor calibration data; we apply it via the LoNAS supernet in §4.3.

B ψ is not parameter count: extended evidence

The main paper (§3.1) claims that spectral capacity ψ is sensitive to architectural structure that parameter count mn cannot see. We provide four independent lines of evidence: (a) controlled rank ablation at fixed dimensions, (b) aspect-ratio dependence at fixed parameter count, (c) heterogeneous operations at matched dimensions, and (d) compression scenarios at fixed base architecture. Throughout, ψ is well-defined on rank-deficient matrices since zeros contribute zero to the sum—a property required by the LLaMA pruning of §4.3 and the rank-truncation analysis below.

(a) Low-rank truncation: same dimensions, same parameter count, different ψ. The cleanest test of $\psi \neq mn$ is to fix the dimensions (m, n) and vary only the spectrum. We take a 4096×11008 `gate_proj` matrix from layer 0 of LLaMA-7B and apply rank-k SVD truncation for $k \in \{1, 2, 4, \ldots, 4096\}$. Each truncated matrix occupies the same $\mathbb{R}^{4096\times11008}$ ambient space, so $mn = 45{,}088{,}768$ is constant across all truncations. Yet ψ varies by more than three orders of magnitude (Figure 4): full rank yields $\psi(W) \approx 3174$, rank 1000 retains only 42.5% of the full capacity ($\psi \approx 1349$), and rank 1 collapses to $\psi \approx 2.5$—a $1{,}270\times$ reduction. The reduction is monotone in k and tracks reconstruction fidelity $\|W - W_k\|_F / \|W\|_F$ closely, confirming that ψ measures the information content of the spectrum rather than the size of the bounding box.

(b) Aspect ratio: same parameter count, different ψ. Even at fixed parameter count $mn = k$, the value of ψ depends on the aspect ratio $\gamma = \min(m, n)/\max(m, n)$. Using the closed form $\psi_{\mathrm{MP}}(m, n, s)$ (Eq. 4, §3.2), we compare matrices analytically: for fixed k and initialization variance

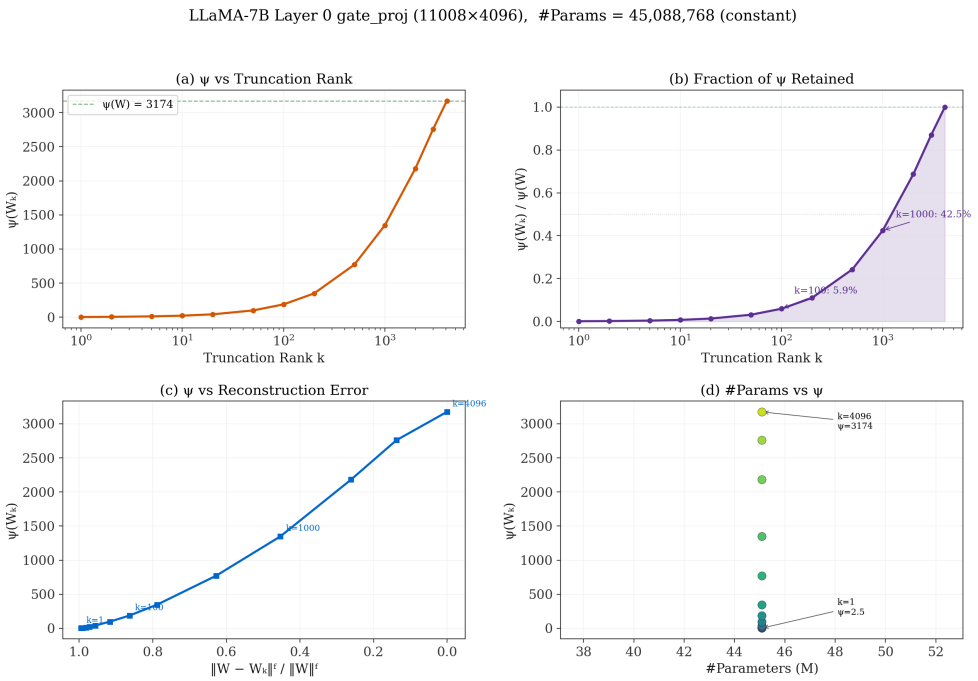


Figure 4: $\psi \neq$ parameter count. Low-rank truncation of the 4096×11008 `gate_proj` matrix from LLaMA-7B layer 0. (a) ψ grows monotonically with truncation rank k. (b) Even at rank 1000, only 42.5% of full capacity is retained. (c) ψ tracks reconstruction fidelity. (d) Parameter count is constant (≈ 45M) across all ranks while ψ varies by orders of magnitude.

s^2, the function $m \mapsto \psi_{\text{MP}}(m, k/m, s)$ is maximized at $m = \sqrt{k}$, i.e., among all rectangular matrices with k parameters, the square matrix has the highest spectral capacity. A geometric intuition for this fact is provided in Appendix D.3.

A concrete consequence: at $k = 4 \times 10^6$ parameters with Xavier initialization ($s^2 = 2/(m+n)$), a 2000×2000 square matrix has $\psi_{\text{MP}} \approx 1{,}161$ (versus a Jensen upper bound $N \ln(1 + Ms^2) = 2000 \ln 2 \approx 1{,}386$), while a 4000×1000 rectangular matrix has $\psi_{\text{MP}} \approx 909$ (versus a Jensen upper bound $1000 \ln(1 + 1.6) \approx 956$)—a $\approx 28\%$ gap with identical mn. Both values are computed by 1D Gauss–Kronrod quadrature of the MP integral and verified against the Tulino–Verdú closed form (Appendix D.4); they sit strictly below their respective Jensen upper bounds, consistent with random-Gaussian rather than perfectly orthogonal spectra. Parameter count cannot distinguish these architectures; ψ_{MP} ranks the square one higher, consistent with the empirical observation that balanced layer dimensions tend to yield better-trained models.

(c) Heterogeneous operations: same dimensions, different ψ. On the FlexiBERT benchmark, layers contain different operation types—self-attention (SA), linear transform (LT), depthwise-separable convolution (DSC)—instantiated with identical hidden size h and FFN width. Parameter count assigns the same score to layers with the same (h, FFN) regardless of operation type, since mn is determined by dimensions alone. ψ, in contrast, distinguishes operation types because each op type contributes a different set of weight matrices to the layer-wise sum in Eq. 5: SA contributes Q/K/V projections plus an output projection; LT contributes a single linear map; DSC contributes a depthwise and a pointwise tensor. The aggregate ψ values differ because the spectral capacity of several smaller matrices is not equal to the spectral capacity of one large matrix at the same total parameter count.

Operation-type structure is the source of FlexiBERT's ranking gap between ψ-based proxies and parameter-counting baselines. Empirically, on the 500-architecture spec-faithful FlexiBERT set (§4), `NSC` achieves Spearman $\rho = 0.884$, while #Params achieves $\rho = 0.676$—a 0.21-point gap that arises principally from ψ's sensitivity to operation-type structure that #Params is blind to.

(d) Compression scenarios: same base architecture, different subnet ψ. A second setting where parameter count fails entirely is structured pruning of a fixed base model. In the LoNAS experiment (§4.3), all subnets share the same LLaMA-7B base architecture: same depth (32 layers), same $d_{\text{model}} = 4096$, same attention configuration. Subnets differ only in which per-layer FFN dimensions are retained from the base $d_{\text{ffn}} = 11008$. At a fixed compression budget (e.g., 5.7B parameters), parameter count is constant by construction across all candidate subnets and provides zero ranking signal. ψ, by contrast, evaluates the spectral content of each pruned weight matrix and ranks subnets by how much spectral capacity the chosen FFN dimensions retain. This is precisely the regime where ψ's structural sensitivity matters most, and it explains why `NSC`-guided selection achieves the highest average across eight commonsense reasoning benchmarks at the 5.7B budget (Table 5).

Summary. Across all four regimes—rank ablation, aspect ratio, heterogeneous operations, and compression—ψ discriminates between architectures or weight matrices that parameter count cannot. The first two are controlled mathematical settings in which mn is held exactly constant; the latter two are practical NAS scenarios where the constraint structure makes #Params uninformative by design. Together, these establish that ψ extracts architectural information genuinely orthogonal to model size.

C Choice of aggregation: protocol and analysis

Protocol. Table 1 (§3.3) compares the across-layer summation of `NSC` against the two natural alternatives; this appendix details the protocol and explains the results. We evaluate on the three ranking benchmarks of §4.1 under their standard protocols: 500 FlexiBERT architectures against GLUE (spec-faithful setting, §F.1), 200 GPT-2 architectures against WikiText-103 perplexity (negated, so that larger ρ is better), and 1,001 AutoFormer-Tiny architectures against ImageNet Top-1. Per-matrix capacities and layer capacities Ψ_l are computed once per architecture following §3.3; the rules differ only in how the Ψ_l are combined: summation, the serial bottleneck $\min_l \Psi_l$—the series half of the composition rule for channel capacities (Foggo and Yu, 2023)—and the product $\prod_l \Psi_l$. Alongside Spearman ρ, Table 1 reports the number of distinct score values each rule assigns: a rule that cannot separate architectures cannot rank them.

Why the bottleneck collapses. The failure of $\min_l \Psi_l$ is structural rather than incidental. Its score always equals one layer's capacity, so a search space admits at most D distinct scores, where D is the number of distinct layer configurations the space allows: $D = 2 \times 3 \times 4 = 24$ for FlexiBERT (hidden size, operation type, FFN configuration), $D = 2 \times 2 \times 2 = 8$ for AutoFormer-Tiny, and $D = 17{,}370$ for GPT-2, whose per-layer FFN width is drawn from a wide integer range. The measured counts (22, 3, 195) sit below these bounds because the minimum concentrates on the weakest configurations: an architecture's minimum avoids a weak configuration only if every layer does—unlikely at depth—so on AutoFormer-Tiny 895 of the 1,001 architectures share a single score and only 3 of the 8 configurations ever serve as the minimum. The same mechanism produces the negative correlation there: adding layers can only lower a minimum, while deeper architectures tend to be more accurate in that space. GPT-2 is the complementary case: D far exceeds the number of architectures, the bound never binds, and summation still outranks the bottleneck (0.968 vs. 0.813).

Why the product conflates depth with capacity. With layer capacities well above one, $\prod_l \Psi_l$ grows geometrically in depth, so differences in L overwhelm differences within layers. The rule is competitive on FlexiBERT (0.917), where depth takes only two values, but drops to 0.712 on GPT-2, where depth varies freely.

Summary. The bottleneck fails by construction on spaces with few layer configurations, and the product conflates depth with capacity; summation avoids both failure modes and is the only additive rule—the structural property `NSC-DP` requires (§3.4).

D Proofs

D.1 Proof of the mutual information identity

We prove the mutual information identity $I(x; y) = \frac{1}{2}\psi(W)$ stated inline in §3.1.

For $y = Wx + z$ with $x \sim \mathcal{N}(0, I_n)$, $z \sim \mathcal{N}(0, I_m)$ independent, we have $y \sim \mathcal{N}(0, WW^\top + I_m)$ and $y \mid x \sim \mathcal{N}(Wx, I_m)$. Using the differential entropy of $\mathcal{N}(\mu, \Sigma)$ on $\mathbb{R}^d$, $H = \frac{1}{2} \ln \det(2\pi e\, \Sigma)$:

$$\begin{aligned} I(x; y) &= H(y) - H(y \mid x) \\ &= \tfrac{1}{2} \ln \det(2\pi e\, (I_m + WW^\top)) - \tfrac{1}{2} \ln \det(2\pi e\, I_m) \\ &= \tfrac{1}{2} \ln \det(I_m + WW^\top) \;=\; \tfrac{1}{2} \ln \det(I_n + W^\top W) \;=\; \tfrac{1}{2} \psi(W), \end{aligned}$$

where the third equality uses $\ln \det(2\pi e\, A) = m \ln(2\pi e) + \ln \det(A)$ for $A \in \mathbb{R}^{m \times m}$ (the $m \ln(2\pi e)$ contribution cancels in the difference), the fourth equality uses Sylvester's determinant identity $\det(I_m + AB) = \det(I_n + BA)$, and the final equality uses $\ln \det(I + W^\top W) = \sum_i \ln(1 + \sigma_i^2)$, valid because the eigenvalues of $W^\top W$ are the squared singular values of W. □

D.2 Gradient Responsiveness

Proposition 1 (Gradient Responsiveness)**.** $\nabla_W \psi(W) = 2W(I + W^\top W)^{-1}$. *In the SVD basis, the per-singular-value gradient weight is $g(\sigma) = 2\sigma/(1 + \sigma^2)$, which attains its unique maximum at $\sigma = 1$.*

Proof. Let $A = W^\top W$. Then $\mathrm{d}\psi = \mathrm{tr}[(I+A)^{-1}\, \mathrm{d}A]$ with $\mathrm{d}A = \mathrm{d}W^\top W + W^\top \mathrm{d}W$. Expanding and using cyclicity of trace:

$$\mathrm{d}\psi = 2\, \mathrm{tr}[(I+A)^{-1} W^\top\, \mathrm{d}W]$$

giving $\nabla_W \psi = 2W(I+W^\top W)^{-1}$. In the SVD basis $W = U\Sigma V^\top$:

$$\nabla_W \psi = U\, \mathrm{diag}\left(\frac{2\sigma_i}{1+\sigma_i^2} \right) V^\top$$

so the per-singular-value gradient weight is $g(\sigma) = 2\sigma/(1+\sigma^2)$. Setting $g'(\sigma) = 2(1-\sigma^2)/(1+\sigma^2)^2 = 0$ gives the unique maximum at $\sigma = 1$. □

Remark. This is the gradient of ψ with respect to its argument W, not the gradient of any task loss. During training, W is updated by $\nabla_W \mathcal{L}_{\text{task}}$, which has no direct relationship to $\nabla_W \psi$. What Proposition 1 establishes is that ψ is most sensitive to singular values of order one—the regime where standard variance-preserving initialization (Xavier, Kaiming) places them. The connection between ψ-dynamics and task-loss dynamics in Figure 1 is empirical, not derived from this gradient analysis.

D.3 Proof of Theorem 1 (Marchenko-Pastur Spectral Capacity)

Let $W \in \mathbb{R}^{m \times n}$ have i.i.d. entries with mean 0 and variance s^2, along a sequence of sizes with $\min(m, n) \to \infty$ and γ and Ms^2 held fixed, as in the statement of the theorem. Without loss of generality assume $m \geq n$ (otherwise consider $W^\top$, which has identical singular values), so that $M = m$, $N = n$.

Step 1: Marchenko-Pastur law. The normalized Gram matrix $S = \frac{1}{Ms^2} W^\top W \in \mathbb{R}^{N \times N}$ has eigenvalues $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_N \geq 0$. By the Marchenko-Pastur theorem Marčenko and Pastur (1967), the empirical spectral distribution

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\tilde{\lambda}_i}$$

converges almost surely (in distribution) to the Marchenko-Pastur law μ_γ with density:

$$f_{\mathrm{MP}}(\lambda; \gamma) = \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\gamma\lambda}, \quad \lambda \in [\lambda_-, \lambda_+]$$

where $\lambda_\pm = (1 \pm \sqrt{\gamma})^2$.

Step 2: From eigenvalues to ψ. The eigenvalues of $W^\top W$ are $Ms^2 \tilde{\lambda}_i$. Since $\psi(W) = \sum_i \ln(1 + \lambda_i(W^\top W))$:

$$\frac{1}{N} \psi(W) = \frac{1}{N} \sum_{i=1}^{N} \ln\!\left(1 + Ms^2 \tilde{\lambda}_i\right) = \int \ln\!\big(1 + Ms^2 \lambda\big)\, \mathrm{d}\mu_N(\lambda)$$

Step 3: Convergence. The integrand $g(\lambda) = \ln(1 + Ms^2\lambda)$ is a fixed function along the sequence, since Ms^2 is held fixed. The extreme nonzero eigenvalues of S converge a.s. to the support edges $\lambda_\pm$ (Bai and Silverstein, 2010), so the spectral mass is eventually contained in a fixed compact set on which g is bounded and continuous, and the weak convergence $\mu_N \to \mu_\gamma$ (a.s.) gives

$$\frac{1}{N}\psi(W) = \int g \, \mathrm{d}\mu_N \xrightarrow{\text{a.s.}} \int_{\lambda_-}^{\lambda_+} \ln\big(1 + Ms^2\lambda\big)\, f_{\mathrm{MP}}(\lambda;\gamma)\, d\lambda \;=\; \mathcal{I}(\gamma, Ms^2),$$

which is the claimed limit. □

Note on finite-size convergence. Under the convention adopted in §3.2, ψ_{MP} is the *definition* of ψ for use as an architecture proxy, not an approximation of a finite-sample quantity. Empirical demonstration of finite-size convergence at typical Transformer dimensions is given in Appendix G.

Geometric intuition for the aspect-ratio result. Appendix B (b) shows numerically that for fixed mn, the square matrix maximizes ψ_{MP}. Intuitively, varying γ at fixed mn trades off the number of eigenvalues $N = \sqrt{\gamma\, mn}$ (favoring $\gamma = 1$) against the per-eigenvalue contribution $\ln(1 + Ms^2\lambda)$ where $M = \sqrt{mn/\gamma}$ scales as $1/\sqrt{\gamma}$ (favoring $\gamma \to 0$). The product $\psi_{\mathrm{MP}}(\gamma) = N\,\mathbb{E}_\lambda[\ln(1 + Ms^2\lambda)]$ is maximized at $\gamma = 1$, the spectral-theoretic counterpart of the classical result that mutual information across parallel Gaussian channels at fixed total power is maximized by uniform power allocation.

D.4 Strict Concavity of ψ_{MP} and Optimality of Uniform FFN Allocation

This appendix substantiates the structural claim made in Section 4.2: under a fixed total feed-forward parameter budget, the NSC-maximizing per-layer FFN allocation is uniform. The role of concavity should be stated precisely. It underwrites the uniform-allocation corollary at the end of this appendix—it explains *what* NSC-DP returns on this space. NSC-DP's global-optimality guarantee does not depend on it: that guarantee follows from layer-wise additivity, exhaustive enumeration of the network-level configurations, and exact solution of the inner knapsack (§3.4), and holds for arbitrary per-layer value tables. Were ψ_{MP} not concave, NSC-DP would still return the global proxy-maximizer—just not necessarily a uniform one.

Proposition 2 (Strict concavity of ψ_{MP} in its dimension arguments)**.** *For fixed m and $s > 0$, the map $n \mapsto \psi_{\mathrm{MP}}(n, m, s)$ is strictly concave on $(0, \infty)$—hence strictly concave along the integer grids NSC-DP searches—and, by the symmetry of ψ_{MP} in its two dimension arguments, the same holds in m. No relation between m and n is assumed.*

In the Transformer-XL instantiation, n plays d_{ff} and m plays d_{model}; the proposition holds on all of $(0, \infty)$, with no restriction to $d_{\mathrm{ff}} \geq d_{\mathrm{model}}$.

Proof. Let $\beta \coloneqq s^2$ and write $M = \max(m, n)$, $N = \min(m, n)$, $\gamma = N/M$ as in Theorem 1.

Step 1: fixed-point representation of ψ_{MP}. Consider the pair of equations

$$\delta \;=\; \frac{1}{1 + \beta n\tilde{\delta}}, \qquad \tilde{\delta} \;=\; \frac{1}{1 + \beta m\delta}. \tag{6}$$

Substituting the second equation into the first shows that a solution must satisfy $G(\delta; n) = 0$, where

$$G(\delta; n) \;\coloneqq\; \delta\left(1 + \frac{\beta n}{1 + \beta m\delta}\right) - 1.$$

Since $G(0; n) = -1 < 0$, $\partial G/\partial\delta = 1 + \beta n/(1 + \beta m\delta)^2 > 0$, and $G(\delta; n) \to \infty$ as $\delta \to \infty$, Eq. (6) has a unique positive solution $(\delta^*, \tilde{\delta}^*)$ for every $m, n, \beta > 0$. The pair is the classical Marchenko–Pastur self-consistency in symmetric form: writing $\eta(x) \coloneqq \int f_{\mathrm{MP}}(\lambda;\gamma)/(1 + x\lambda)\, d\lambda$ for the η-transform of the MP law, the unique positive solution is

$$\delta^* = \frac{m - N + N\,\eta(M\beta)}{m}, \qquad \tilde{\delta}^* = \frac{n - N + N\,\eta(M\beta)}{n}, \tag{7}$$

the deterministic equivalents of the normalized resolvent traces $\frac{1}{m}\operatorname{tr}(I_m + W^\top W)^{-1}$ and $\frac{1}{n}\operatorname{tr}(I_n + WW^\top)^{-1}$ for $W \in \mathbb{R}^{n\times m}$ with i.i.d. entries of variance s^2 (Tulino and Verdú, 2004, Thm. 2.39).

Define

$$\Phi(\delta, \tilde{\delta}; n) \;\coloneqq\; n\ln(1+\beta m\delta) \;+\; m\ln(1+\beta n\tilde{\delta}) \;-\; \beta mn\,\delta\tilde{\delta}. \tag{8}$$

Direct differentiation gives $\partial\Phi/\partial\delta = \beta mn\left[(1+\beta m\delta)^{-1} - \tilde{\delta}\right]$ and $\partial\Phi/\partial\tilde{\delta} = \beta mn\left[(1+\beta n\tilde{\delta})^{-1} - \delta\right]$, so $(\delta, \tilde{\delta})$ is a stationary point of $\Phi(\cdot,\cdot\,; n)$ exactly when it solves Eq. (6). We claim the value identity

$$\psi_{\mathrm{MP}}(n, m, s) \;=\; \Phi(\delta^*, \tilde{\delta}^*; n). \tag{9}$$

To derive it, view both sides as functions of β at fixed (m, n), with $(\delta^*(\beta), \tilde{\delta}^*(\beta))$ the positive root of Eq. (6). At $\beta = 0$ both sides vanish (there $\delta^* = \tilde{\delta}^* = 1$). For the left side, differentiating the MP integral (Eqs. 3 and 4) in β gives

$$\frac{d\psi_{\mathrm{MP}}}{d\beta} = N\int \frac{M\lambda}{1+\beta M\lambda}\, f_{\mathrm{MP}}(\lambda;\gamma)\, d\lambda = \frac{N\big(1-\eta(M\beta)\big)}{\beta} = \frac{n\big(1-\tilde{\delta}^*\big)}{\beta} = mn\,\delta^*\tilde{\delta}^*,$$

where the third equality uses Eq. (7) and the fourth uses $1-\tilde{\delta}^* = \beta m\delta^*\tilde{\delta}^*$, a rearrangement of the second fixed-point equation. For the right side, stationarity means only the explicit β-dependence contributes (envelope theorem):

$$\frac{d}{d\beta}\,\Phi(\delta^*, \tilde{\delta}^*; n) = \left.\frac{\partial\Phi}{\partial\beta}\right|_{(\delta^*,\,\tilde{\delta}^*)} = \frac{mn\,\delta^*}{1+\beta m\delta^*} + \frac{mn\,\tilde{\delta}^*}{1+\beta n\tilde{\delta}^*} - mn\,\delta^*\tilde{\delta}^* = mn\,\delta^*\tilde{\delta}^*,$$

applying the two fixed-point equations to the first two terms. Equal values at $\beta = 0$ and equal β-derivatives establish Eq. (9). (Numerically, the two sides of Eq. (9) agree to relative error below 10^{-15} across the dimension ranges of our benchmarks.)

Step 2: envelope identity for the marginal capacity. Differentiating Eq. (9) in n, stationarity again cancels the dependence through $(\delta^*, \tilde{\delta}^*)$:

$$\frac{d\psi_{\mathrm{MP}}}{dn} = \left.\frac{\partial\Phi}{\partial n}\right|_{(\delta^*,\,\tilde{\delta}^*)} = \ln\big(1+\beta m\delta^*\big) + \frac{\beta m\tilde{\delta}^*}{1+\beta n\tilde{\delta}^*} - \beta m\,\delta^*\tilde{\delta}^* = \ln\big(1+\beta m\,\delta^*(n)\big), \tag{10}$$

the last two terms cancelling by the first fixed-point equation.

Step 3: monotonicity and strict concavity. Implicit differentiation of $G(\delta^*(n); n) = 0$ gives

$$\frac{d\delta^*}{dn} \;=\; -\,\frac{\partial G/\partial n}{\partial G/\partial\delta} \;=\; -\,\frac{\beta\delta^*/(1+\beta m\delta^*)}{1+\beta n/(1+\beta m\delta^*)^2} \;<\; 0,$$

hence

$$\frac{d^2\psi_{\mathrm{MP}}}{dn^2} \;=\; \frac{\beta m}{1+\beta m\,\delta^*(n)}\,\frac{d\delta^*}{dn} \;<\; 0 \qquad \text{for all } n \in (0,\infty).$$

Strict concavity in m follows from the invariance of Φ under the simultaneous swap $(n,\delta) \leftrightarrow (m,\tilde{\delta})$, mirroring the symmetry $\psi_{\mathrm{MP}}(n, m, s) = \psi_{\mathrm{MP}}(m, n, s)$. □

Remark: the envelope formula is the rank-one mechanism, made deterministic. Adding one row $w \sim \mathcal{N}(0, s^2 I_m)$ to W changes ψ by $\ln(1 + w^\top(I + W^\top W)^{-1}w)$ (matrix determinant lemma), whose conditional expectation concentrates on $\ln(1 + s^2\,\mathrm{tr}\,(I + W^\top W)^{-1}) \approx \ln(1+\beta m\delta^*)$—exactly Eq. (10). The probabilistic argument below runs on this mechanism at finite sizes; the proof above is its deterministic sharpening.

A second, probabilistic proof of (non-strict) discrete concavity. Let $W_d \in \mathbb{R}^{d\times m}$ have i.i.d. $\mathcal{N}(0, s^2)$ entries, and let $M_d := I + W_d^\top W_d$. Adding one row $w \sim \mathcal{N}(0, s^2 I_m)$ independent of W_d gives $M_{d+1} = M_d + ww^\top$. By the matrix determinant lemma,

$$\psi(W_{d+1}) - \psi(W_d) \;=\; \ln\det M_{d+1} - \ln\det M_d \;=\; \ln\big(1 + w^\top M_d^{-1} w\big).$$

Taking conditional expectation over w (write $w = s\,z$, $z \sim \mathcal{N}(0, I_m)$),

$$\mathbb{E}\big[\psi(W_{d+1}) - \psi(W_d) \;\big|\; W_d\big] \;=\; \mathbb{E}_z\left[\ln\Big(1 + s^2\sum_{i=1}^{m}\tfrac{z_i^2}{1+\sigma_i^2(W_d)}\Big)\right],$$

where $\sigma_i^2(W_d)$ are the eigenvalues of $W_d^\top W_d$. As d increases, each $\sigma_i^2(W_d)$ grows (rank-1 PSD updates can only increase eigenvalues, by Weyl's inequality), so each coefficient $1/(1+\sigma_i^2(W_d))$ shrinks. Since $\ln(1+\cdot)$ is monotone increasing, the expected marginal gain $\mathbb{E}[\psi(W_{d+1}) - \psi(W_d)]$ is non-increasing in d: the map $d \mapsto \mathbb{E}[\psi(W_d)]$ is discretely concave at every finite size.

Passing from this statement to the concavity of ψ_{MP} requires care: by the central limit theorem for linear spectral statistics (Bai and Silverstein, 2010), $\psi(W) - \psi_{\mathrm{MP}}$ is $O_P(1)$, not $o(1)$—only the relative deviation vanishes (Appendix G.1)—so a limit taken along n cannot be exchanged with a second difference in n. The passage is completed by taking the limit along the family that realizes ψ_{MP} at *finite* (m, n). Because ψ_{MP} depends on the specification only through (N, γ, Ms^2), it is exactly 1-homogeneous under the replica map $(n, m, s) \mapsto (kn, km, s/\sqrt{k})$, which fixes both γ and Ms^2: $\psi_{\mathrm{MP}}(kn, km, s/\sqrt{k}) = k\,\psi_{\mathrm{MP}}(n, m, s)$ for every $k \in \mathbb{N}$. Along this family the hypotheses of Theorem 1 hold with γ and Ms^2 *fixed*, so

$$\tfrac{1}{k}\,\psi\big(W^{(k)}\big) \;=\; N \cdot \frac{\psi(W^{(k)})}{kN} \;\xrightarrow{\text{a.s.}}\; N\,\mathcal{I}(\gamma, Ms^2) \;=\; \psi_{\mathrm{MP}}(n, m, s) \qquad (k \to \infty),$$

with $W^{(k)}$ of size $kn \times km$ and i.i.d. entries of variance s^2/k. Uniform integrability upgrades this to convergence of expectations: $0 \le \psi(W) = \ln\det(I + W^\top W) \le \mathrm{tr}(W^\top W) = \|W\|_F^2$, and $\frac{1}{k}\|W^{(k)}\|_F^2$ is an average of i.i.d. variables with expectation mns^2 for every k, converging a.s. to that constant, so Pratt's generalized dominated convergence theorem gives $\mathbb{E}\big[\frac{1}{k}\psi(W^{(k)})\big] \to \psi_{\mathrm{MP}}(n, m, s)$. The rank-one argument applies unchanged at each k: incrementing n by one adds k rows at fixed column count and entry variance, and a sum of k consecutive non-increasing marginal gains is non-increasing, so every $n \mapsto \mathbb{E}\big[\frac{1}{k}\psi(W^{(k)})\big]$ is discretely concave. Discrete concavity at a grid point is a single closed inequality among three values, so it passes to the pointwise limit: $n \mapsto \psi_{\mathrm{MP}}(n, m, s)$ is discretely concave. □

Numerical check of the implementation. As an independent check that the quadrature implementation of ψ_{MP} exhibits the concavity proven in Proposition 2, Figure 5 plots the marginal gain $\partial\psi_{\mathrm{MP}}/\partial d_{\mathrm{ff}}$ across all $(d_{\mathrm{model}}, d_{\mathrm{ff}})$ pairs in our Transformer-XL search space: all seven curves are strictly decreasing, and all 189 discrete second differences are strictly negative. Table 7 reports the marginal gain at the two endpoints of the search range: it decreases by roughly $2\times$ for every value of d_{model}, consistent with the proposition.

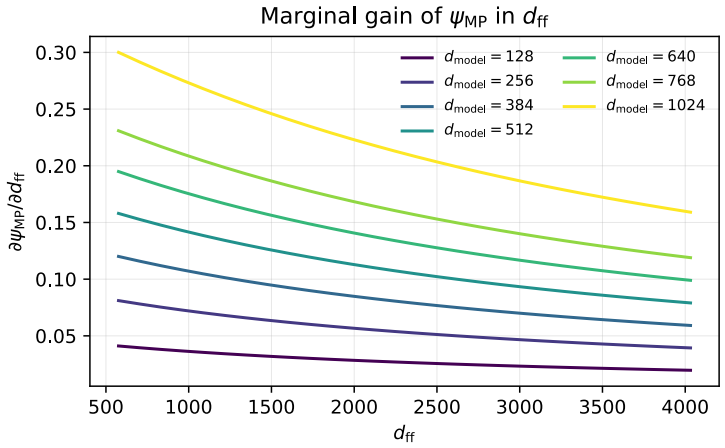


Figure 5: **Implementation check of Proposition 2.** Marginal gain $\partial\psi_{\mathrm{MP}}/\partial d_{\mathrm{ff}}$ (forward differences over a d_{ff} grid of step 128) as a function of d_{ff} for each of the seven embedding dimensions in the Transformer-XL search space, computed at $s = 0.02$. All seven curves are strictly decreasing, with all 189 second differences of $\psi_{\mathrm{MP}}(d_{\mathrm{ff}}, d_{\mathrm{model}}, s)$ negative across the full range, as the proposition requires.

Corollary (Uniform allocation optimality). Under a fixed total FFN parameter budget $\sum_{l=1}^{L} d_{\mathrm{ff},l} \cdot (2d_{\mathrm{model}} + 1) \le B$, the unique NSC-maximizing per-layer allocation $\{d_{\mathrm{ff},l}\}_{l=1}^{L}$ of the continuous re-

Table 7: Marginal gain $\partial\psi_{\mathrm{MP}}/\partial d_{\mathrm{ff}}$ at the two endpoints of the FFN search range, for each embedding dimension. The marginal gain attenuates by roughly $2\times$ across the range, consistent with Proposition 2.

d_{model}	at $d_{\mathrm{ff}} = 512$	at $d_{\mathrm{ff}} = 4096$	all 2nd diff. < 0?
128	0.041	0.020	✓
256	0.081	0.040	✓
384	0.120	0.060	✓
512	0.158	0.080	✓
640	0.195	0.100	✓
768	0.231	0.120	✓
1024	0.300	0.161	✓

laxation is uniform, $d_{\mathrm{ff},l} = B/[L(2d_{\mathrm{model}}+1)]$ for all l; on the discrete search grid the optimum is its rounding. This is Jensen's inequality applied to Proposition 2: the objective $\sum_l \psi_{\mathrm{MP}}(d_{\mathrm{ff},l}, d_{\mathrm{model}}, s)$ is a sum of strictly concave functions of the $d_{\mathrm{ff},l}$ under a linear constraint on $\sum_l d_{\mathrm{ff},l}$, so replacing any non-uniform allocation by its average strictly increases it. The uniform allocation discovered by `NSC-DP` in Section 4.2 is therefore not coincidental but a structural consequence of `NSC`'s spectral form. □

E Multi-Budget Search via `NSC-DP`

E.1 Motivation

We elaborate on the multi-budget property stated in §3.4. Because (i) the resource axis (parameters or, equivalently in the LoNAS architecture, TFLOPs) is an integer sum of per-block contributions and (ii) the spectral-capacity score ψ_{MP} is per-layer additive, the DP table after the final layer encodes

$$\mathcal{P}_{\texttt{NSC-DP}} \;=\; \Big\{\big(b,\ \max_{\text{subnet}\,:\,\textsc{Params}\le b}\ \textstyle\sum_\ell \Psi_\ell(r_\ell, h_\ell)\big)\ :\ b \in \mathcal{B}\Big\}, \tag{11}$$

where $\Psi_\ell(r_\ell, h_\ell)$ is the layer capacity (§3.3) at the chosen (r_ℓ, h_ℓ) configuration. $\mathcal{P}_{\texttt{NSC-DP}}$ is the entire Pareto front over the discretized budget grid $\mathcal{B}$. Switching the deployment budget therefore requires *neither additional search nor a new proxy cache*; the practitioner reads a different cell of the same DP table.

E.2 Empirical demonstration: seven subnets from a single DP run

To make this property concrete, we instantiate the seven operating points ("Subnet-A" through "Subnet-G") reported in Munoz et al. (2024)'s Table 4 by reading them off a *single* `NSC-DP` execution. No re-search, no per-budget retraining, and no per-budget proxy re-computation is performed: all seven configurations are obtained from the same ψ_{MP} cache and the same DP backward pass used in §4.3. A–G correspond to LoNAS's published TFLOPs operating points, while the main-text result is read at the 5.7 B parameter budget—a different point on the same Pareto front.

Table 8 reports the resulting subnets, their TFLOPs (computed at the conventional $T = 256$ sequence length used throughout the LoNAS literature), per-task accuracy under our unified evaluation protocol, and the amortised search wall-clock time per operating point. The LoNAS-SuperNet (maximal configuration, $r_\ell = 32$, $h_\ell = 11008$ for all ℓ) is included as the upper-bound reference.

Figure 3 (main paper) visualizes the corresponding accuracy–TFLOPs Pareto front; the monotone shape is a structural consequence of the per-layer additivity of ψ_{MP} (adjacent budgets produce adjacent subnets, removing the run-to-run variance of stochastic searches).

E.3 Discussion

Two observations follow from Table 8.

Table 8: **Seven operating points along the `NSC-DP` Pareto front, extracted from a single DP backward pass.** TFLOPs are computed at sequence length $T = 256$. All seven `NSC-DP` rows share *the same* 0.46 s DP backward pass: they are not seven independent searches but seven read-outs of one DP table. The marginal cost of obtaining an additional operating point is therefore *exactly zero*; switching budgets is an $\mathcal{O}(B)$ table look-up. The LoNAS-SuperNet row is the maximal-configuration upper bound and performs no search.

Subnet	**TFLOPs**	BoolQ	PIQA	SIQA	HSwag	WGrnd	ARC-e	ARC-c	OBQA	**Avg$_8$**	**Search**
LoNAS-SuperNet	1.71	66.02	77.58	72.93	57.49	66.85	78.41	62.29	76.20	**69.72**	—
`NSC-DP-A`	1.41	64.01	72.42	68.17	51.26	64.09	72.52	58.53	70.60	65.20	
`NSC-DP-B`	1.40	63.73	72.42	68.27	51.05	63.46	72.18	58.79	71.20	65.14	
`NSC-DP-C`	1.39	63.61	72.25	67.50	50.69	63.38	72.18	58.87	72.00	65.06	**0.46 s**
`NSC-DP-D`	1.38	63.73	72.03	66.73	49.95	63.85	71.34	58.19	70.80	64.58	*(shared,*
`NSC-DP-E`	1.37	63.15	71.49	67.71	49.73	63.69	71.30	57.08	71.00	64.39	*single DP)*
`NSC-DP-F`	1.36	62.51	69.37	66.63	46.54	62.51	68.06	54.86	67.80	62.29	
`NSC-DP-G`	1.29	59.27	69.48	64.64	44.73	62.27	66.04	52.82	66.60	60.73	

(i) The DP table is the Pareto front. The seven `NSC-DP` rows are not seven independent searches: they are seven read-outs of the same 0.46 s DP backward pass. Every *additional* budget the practitioner queries costs $\mathcal{O}(B)$ table look-ups (microseconds), not a new search.

(ii) Smooth, monotone front. Avg_8 decreases monotonically from 65.20% at 1.41 TFLOPs (`NSC-DP-A`) to 60.73% at 1.29 TFLOPs (`NSC-DP-G`)—a 4.47 pp range across 0.12 TFLOPs of compute reduction. The smoothness of this front is itself a downstream consequence of the closed-form, additive structure of ψ_{MP}: small changes in the budget produce small, predictable changes in the recovered subnet, removing the run-to-run variance that plagues stochastic searches.

Summary. `NSC-DP` returns Pareto-front output as a *free* structural consequence of its closed-form, additive formulation rather than as the output of an explicit per-budget search. Among the methods evaluated in this paper, `NSC-DP` is the only one that returns an entire Pareto front in a single sub-second invocation while requiring neither labels nor GPU forward passes.

F Extended Experimental Results

F.1 FlexiBERT: original (Serianni) vs. spec-faithful setting

The 500-architecture FlexiBERT benchmark used in §4.1 originates from Tuli et al. (2023), who defined a heterogeneous Transformer search space in which each architecture independently varies four dimensions: hidden size $h \in \{128, 256\}$, depth $L \in \{2, 4\}$, *per-layer* feed-forward dimension $d_{\text{ff},l} \in \{512, 1024\}$, and per-layer attention operation type. Serianni and Kalita (2023) subsequently retrained these architectures on GLUE to obtain ground-truth scores, and this is the standard ranking benchmark adopted by all subsequent training-free Transformer proxies (W-PCA (Wang, 2025), ZeroLM (Chen et al., 2025), attention-based proxies in Serianni and Kalita (2023)).

The issue with the original (Serianni) evaluation protocol. The proxy-evaluation code released by Serianni and Kalita (2023), and reused by subsequent training-free proxies that report numbers on this benchmark, instantiates *every* candidate architecture with two of the four search dimensions hard-coded to fixed constants. Concretely, regardless of the architecture's declared h and $\{d_{\text{ff},l}\}$, the model is built with `hidden_size` $= 256$ and `intermediate_size` $= 1024$ for all layers (see the original `build_flexibert` implementation; the same constants are reproduced whenever subsequent proxy papers reuse `build_flexibert` as the model factory). This collapses two of the four search dimensions: in our census of the 500-architecture set, $251/500$ architectures declare $h = 128$ in their specification but are scored at $h = 256$, and the per-layer FFN structure—one of the two dimensions FlexiBERT was specifically designed to expose—is replaced by a constant. The downstream effect is that any proxy whose value depends on the architecture specification (parameter count, FLOPs, W-PCA, `NSC`, etc.) is computed on a *scrambled* spec rather than the architecture whose GLUE score is being predicted. Proxies that score from instantiated weights or activations

are similarly affected, since the model they instantiate no longer matches the architecture that was trained.

The spec-faithful setting. We restore the search-space-faithful protocol: each of the 500 architectures is scored with its declared $(h, \{d_{\text{ff},l}\})$, recovering the full $\{128, 256\} \times \{512, 1024\}^L$ structure across the 500 specs. The set of architectures, the GLUE ground truth, and the proxy implementations are otherwise identical to the original setting. We refer to this as the *Spec-faithful* setting and use it as the default in §4.1; results under the original *Serianni* setting are reported below for direct comparability with prior tables.

Empirical effect: structure-aware proxies improve, structure-blind ones do not. Table 9 reports Kendall τ and Spearman ρ against ground-truth GLUE for representative proxies under both settings. Two observations: *(i)* Architectural quantities improve under the spec-faithful protocol, with NSC gaining substantially (τ: $0.544 \rightarrow 0.695$) and #Params modestly ($0.454 \rightarrow 0.485$). *(ii)* Among training-free proxies, responses diverge: W-PCA, SNIP, GradNorm, and SynFlow recover from suppressed values (W-PCA: $0.522 \rightarrow 0.638$; SynFlow: $-0.107 \rightarrow 0.028$), consistent with their reported numbers being evaluated on a population whose two of four declared dimensions are collapsed; ZeroLM, conversely, *decreases* ($0.543 \rightarrow 0.527$), suggesting its prior lead drew partly on the collapsed evaluation rather than on architectural signal. Together these patterns confirm that the original protocol systematically depresses the measurable signal of structure-sensitive proxies and that the spec-faithful setting, not Serianni, is the appropriate apples-to-apples ranking benchmark for the FlexiBERT search space as originally defined.

Table 9: **FlexiBERT ranking quality: Serianni vs. spec-faithful protocol.** Same 500 architectures and GLUE ground truth; only the architecture spec passed to the proxy differs. Architectural quantities (NSC, #Params) improve under the spec-faithful protocol; among training-free proxies, W-PCA, SNIP, GradNorm, and SynFlow recover from suppressed values while ZeroLM *decreases*. **Bold**: best per column; underline: second best.

	Serianni (original)		**Spec-faithful (ours)**	
Method	τ	ρ	τ	ρ
NSC **(ours)**	**0.544**	**0.776**	**0.695**	**0.884**
#Params	0.454	0.652	0.485	0.676
W-PCA (Wang, 2025)	0.522	0.757	<u>0.638</u>	<u>0.850</u>
ZeroLM (Chen et al., 2025)	<u>0.543</u>	<u>0.775</u>	0.527	0.730
SNIP (Lee et al., 2019)	0.096	0.146	0.275	0.411
GradNorm (Abdelfattah et al., 2021)	0.013	0.025	0.210	0.316
SynFlow (Tanaka et al., 2020)	-0.107	-0.159	0.028	0.040

F.2 Controlled-correlation analysis across architecture families

To test whether NSC contributes ranking signal beyond #Params and #FLOPs, we report two complementary controls on every benchmark in our evaluation: a non-parametric *windowed* Kendall τ and *Kendall partial correlation* (Conover and Iman, 1981; Kendall, 1942).

Windowed control. For each control variable $Z \in \{\#\text{Params}, \#\text{FLOPs}\}$, we restrict τ to architecture pairs (i, j) with $|Z_i - Z_j| / \max(Z_i, Z_j) < 10\%$, removing Z as a discriminating signal.

Kendall partial correlation (Conover–Iman, used uniformly). Windowed control is non-parametric but uses only a sample-sparse subset ($\sim$10% of pairs) and depends on the threshold. As a complementary, threshold-free analysis we compute Kendall partial correlation $\tilde{\tau}$ via the Conover–Iman (Conover and Iman, 1981) rank-OLS estimator: rank-transform all signals, OLS-residualize the proxy and ground-truth ranks against the controlled ranks, and apply Kendall τ to the residuals. We use this estimator uniformly for both single-variable controls ($Z = \#\text{Params}$ or $\#\text{FLOPs}$) and joint ($\#\text{Params}, \#\text{FLOPs}$) control, ensuring that columns are mathematically comparable and that no estimator-switch artifacts can creep in across columns. (For reference, Kendall's (1942) classical closed-form $\tilde{\tau}_{XY \cdot Z} = (\tau_{XY} - \tau_{XZ}\tau_{YZ}) / \sqrt{(1 - \tau_{XZ}^2)(1 - \tau_{YZ}^2)}$ produces values that differ from

the Conover–Iman estimate by at most ± 0.06 on every entry of Table 10; we report Conover–Iman as the primary number for cross-column consistency.)

Per-benchmark caveats. *GPT-2:* the LiteTransformerSearch GPT-2 search space varies d_{model} and L at fixed FFN ratio, so $\#\text{Params}$ and $\#\text{FLOPs}$ are rank-equivalent across all architectures (their pairwise τ is exactly 1); single-variable Kendall partial controls collapse the two columns to identical values. *CNN benchmarks (NATS-Bench-SSS, MobileNetV3):* ZeroLM is Transformer-specific and therefore not applicable.

FlexiBERT. Under #Params-windowed control, #Params itself collapses to $\tau = 0.082$ (near random) while NSC retains $\tau = 0.505$. Under #FLOPs-windowed control, #FLOPs collapses to $\tau \approx -0.03$ while NSC retains $\tau = 0.298$. Under joint $(\#\text{Params}, \#\text{FLOPs})$ control, #Params and #FLOPs themselves collapse to $\tilde{\tau} \leq 0.06$ as expected, while NSC retains $\tilde{\tau} = 0.477$, exceeding the strongest training-free competitor W-PCA (0.444) by 0.033, with ZeroLM further behind at 0.217. This confirms that NSC captures architectural information beyond #Params and #FLOPs—the structural signal that size and compute miss.

Table 10: **Controlled-correlation analysis on FlexiBERT.** Methods grouped as in the main-text Table 3; FlexiBERT is scored under the spec-faithful protocol of §F.1. *Windowed τ ($<10\%$):* Kendall τ on architecture pairs whose #Params (resp. #FLOPs) differ by less than 10%. *Kendall partial $\tilde{\tau}$:* **Conover–Iman (Conover and Iman, 1981) rank-OLS estimator used uniformly** for single-variable and joint controls (see text). "—" marks self-control (NaN by definition; numerical OLS residuals are reported as ≤ 0.05 noise and treated as undefined). **Bold**: best; underline: second best.

		Windowed τ ($<10\%$)		**Kendall partial $\tilde{\tau}$ (Conover–Iman)**		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	**0.695**	**0.505**	0.298	**0.584**	**0.477**	**0.477**
#Params	0.485	0.082	0.273	—	0.237	0.013
#FLOPs	0.552	0.335	−0.030	0.412	—	0.053
W-PCA	0.635	0.417	**0.341**	0.508	0.474	0.444
ZeroLM	0.527	0.356	0.052	0.412	0.220	0.217
SNIP	0.289	0.237	−0.061	0.258	0.073	0.109
GradNorm	0.171	0.164	−0.021	0.173	0.077	0.100

GPT-2. Under #Params-windowed control NSC retains $\tau = 0.536$, narrowly leading W-PCA (0.531) and tied with ZeroLM (0.536); under #FLOPs-windowed control NSC leads at $\tau = 0.436$. Under joint $(\#\text{Params}, \#\text{FLOPs})$ control all signals collapse the same way (single-variable and joint columns are degenerate; see caveat above) and NSC retains $\tilde{\tau} = 0.329$, behind W-PCA (0.382) and ZeroLM (0.377).

Table 11: **Controlled-correlation analysis on GPT-2 (LiteTransformerSearch).** Same protocol as Table 10.

		Windowed τ ($<10\%$)		**Kendall partial $\tilde{\tau}$ (Conover–Iman)**		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	0.849	**0.536**	**0.436**	0.329	0.329	0.329
#Params	0.854	0.530	0.395	—	—	0.214
#FLOPs	0.854	0.530	0.395	—	—	0.214
W-PCA	**0.878**	0.531	0.428	**0.382**	**0.382**	**0.382**
ZeroLM	0.841	0.536	0.431	0.377	0.377	0.377
SNIP	0.856	0.451	0.312	0.130	0.130	0.130
GradNorm	0.869	0.448	0.311	0.312	0.312	0.312

AutoFormer-Tiny. NSC is the top-ranked method in every column of the table: under #Params-windowed control NSC retains $\tau = 0.503$ vs. 0.322 for #Params; under joint $(\#\text{Params}, \#\text{FLOPs})$ control it retains $\tilde{\tau} = 0.439$, well above the second-best ZeroLM (0.242). Gradient-based proxies (SNIP, GradNorm) are at or near zero correlation under every controlled column.

Table 12: **Controlled-correlation analysis on AutoFormer-Tiny.** Same protocol as Table 10.

		Windowed τ (<10%)		Kendall partial $\tilde{\tau}$ (Conover–Iman)		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	**0.610**	**0.503**	**0.494**	**0.324**	**0.317**	**0.439**
#Params	0.466	0.322	0.296	—	0.033	−0.041
#FLOPs	0.512	0.385	0.359	0.187	—	−0.147
W-PCA	0.500	0.365	0.342	0.205	0.132	0.114
ZeroLM	0.506	0.428	0.424	0.251	0.133	0.242
SNIP	0.095	−0.014	−0.028	−0.096	−0.109	−0.070
GradNorm	−0.059	−0.082	−0.090	−0.111	−0.107	−0.080

AutoFormer-Small. The pattern of AutoFormer-Tiny replicates: NSC ranks first in every column, with $\tau = 0.803$ overall, $\tau = 0.423$ under #Params-windowed control, and $\tilde{\tau} = 0.253$ under joint partial control (vs. 0.159 for ZeroLM and $\leq$0.086 for #Params, #FLOPs, and W-PCA).

Table 13: **Controlled-correlation analysis on AutoFormer-Small.** Same protocol as Table 10.

		Windowed τ (<10%)		Kendall partial $\tilde{\tau}$ (Conover–Iman)		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	**0.803**	**0.423**	**0.419**	**0.205**	**0.249**	**0.253**
#Params	0.789	0.366	0.364	—	0.053	0.086
#FLOPs	0.787	0.361	0.359	−0.019	—	−0.039
W-PCA	0.785	0.353	0.352	0.064	0.083	0.062
ZeroLM	0.797	0.406	0.397	0.171	0.190	0.159
SNIP	0.697	0.121	0.132	−0.022	−0.018	−0.022
GradNorm	0.541	0.009	0.022	−0.019	−0.018	−0.019

NATS-Bench-SSS (CIFAR-100). NSC retains the highest correlation in every column of the table despite being applied without modification to convolutional channel widths: $\tau = 0.372$ under #Params-windowed control (vs. 0.116 for #Params), and $\tilde{\tau} = 0.470$ under joint partial control (vs. 0.305 for the next-best W-PCA, and $\leq$0.030 for #Params and #FLOPs).

Table 14: **Controlled-correlation analysis on NATS-Bench-SSS (CIFAR-100).** Same protocol as Table 10.

		Windowed τ (<10%)		Kendall partial $\tilde{\tau}$ (Conover–Iman)		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	**0.706**	**0.372**	**0.767**	**0.377**	**0.765**	**0.470**
#Params	0.653	0.116	0.698	—	0.683	0.028
#FLOPs	0.250	−0.340	0.024	−0.335	—	0.030
W-PCA	0.674	0.254	0.721	0.255	0.715	0.305
SNIP	0.496	0.002	0.486	−0.006	0.493	0.206
GradNorm	0.289	0.005	0.178	0.007	0.186	0.141

MobileNetV3 (ImageNet, r=224). The Once-for-All (OFA) (Cai et al., 2020) supernet exposes 2.0×10^{19} subnets via per-stage depth, per-layer channel expansion ratio, and per-layer kernel size; we use the standard 2040-architecture evaluation subset. NSC treats each depthwise-separable convolution as two reshaped 2D matrices (depthwise $\widetilde{W} \in \mathbb{R}^{c \times k^2}$ and pointwise $\widetilde{W} \in \mathbb{R}^{c_{\text{out}} \times c_{\text{in}}}$) and applies ψ_{MP} to each. NSC again ranks first in four of six columns: $\tau = 0.812$ overall and $\tau = 0.535$ under #Params-windowed control (vs. 0.414 for #Params and 0.436 for the next-best SNIP). Under joint partial control NSC retains $\tilde{\tau} = 0.278$, the highest among methods evaluated, with SNIP second (0.211).

Table 15: **Controlled-correlation analysis on MobileNetV3 r=224 (Once-for-All).** Same protocol as Table 10.

		Windowed τ ($<10\%$)		Kendall partial $\tilde{\tau}$ (Conover–Iman)		
Method	τ	| #P	| #F	| #P	| #F	| #P,#F
NSC **(ours)**	**0.812**	**0.535**	0.665	**0.439**	0.668	**0.278**
#Params	0.780	0.414	**0.668**	—	**0.680**	0.025
#FLOPs	0.610	0.379	0.187	0.392	—	0.011
W-PCA	0.630	0.292	0.400	0.234	0.408	0.137
SNIP	0.687	0.436	0.392	0.434	0.397	0.211
GradNorm	0.577	0.296	0.231	0.304	0.242	0.099

F.3 AutoFormer-Small and AutoFormer-Base end-to-end search

The AutoFormer search space is released in three size tiers (Tiny, Small, Base); the main paper (§4.2, Table 4) reports end-to-end search on AutoFormer-Tiny. Here we extend the same protocol to AutoFormer-Small and AutoFormer-Base.

Setup. For each tier, NSC-DP is instantiated on the multiple-choice knapsack (network-level g is depth and embedding dimension; layer-level x_l is per-layer MLP ratio and head count) and selects the proxy-optimal architecture under the corresponding parameter budget (matching the AutoFormer baseline's parameter count for each tier). Ground-truth top-1 / top-5 accuracies are evaluated using the official AutoFormer supernet for each tier. Baselines match §4.2: the AutoFormer oracle (Chen et al., 2021a), TF-TAS (Zhou et al., 2022), AZ-NAS (Lee and Ham, 2024), and W-PCA (Wang, 2025).

Results. Table 16 reports accuracy and search cost on AutoFormer-Small and AutoFormer-Base. NSC-DP's search completes in 0.345 s (Small) and 0.600 s (Base) on a single CPU core—between 6 and 7 orders of magnitude faster than the AutoFormer oracle (24 GPU-days) and 3 to 5 orders of magnitude faster than every training-free proxy baseline. On Acc@1, NSC-DP places second on Small (81.392, 0.258 points behind the oracle) and third on Base (82.086, 0.008 points behind TF-TAS); on Acc@5, NSC-DP ranks second on Base (95.688, 0.058 points behind the oracle) and third on Small (95.644, 0.07 points behind the oracle).

Ranking quality. Per-method ranking on AutoFormer-Small is reported in Table 13 (§F.2); NSC ranks first across all controlled columns.

Table 16: Architecture search on the AutoFormer-Small and AutoFormer-Base supernets. AutoFormer-Tiny results are reported in main-paper Table 4. Search cost in natural units (d = GPU days, s = seconds); AutoFormer / TF-TAS / AZ-NAS costs are reported by the respective authors, W-PCA and NSC-DP are wall-clock. NSC-DP runs on a single CPU core; baselines run on GPU. **Bold**: best per column within each Setting block; underline: second best.

Setting	Method	#Params (M)	Acc@1	Acc@5	Search
Small	AutoFormer (Chen et al., 2021a)	22.9	**81.650**	**95.712**	24 d
	TF-TAS (Zhou et al., 2022)	22.8	81.344	95.698	0.5 d
	AZ-NAS (Lee and Ham, 2024)	23.0	81.182	95.636	5,184 s
	W-PCA (Wang, 2025)	22.8	81.296	95.628	560.61 s
	NSC-DP **(ours)**	23.0	81.392	95.644	**0.345 s**
Base	AutoFormer (Chen et al., 2021a)	54.0	**82.384**	**95.746**	24 d
	TF-TAS (Zhou et al., 2022)	54.0	82.094	95.652	0.5 d
	AZ-NAS (Lee and Ham, 2024)	53.7	81.914	95.588	9,504 s
	W-PCA (Wang, 2025)	53.9	81.928	95.594	1,001.84 s
	NSC-DP **(ours)**	53.9	82.086	95.688	**0.600 s**

F.4 NATS-Bench-SSS end-to-end search

NATS-Bench distinguishes *topology* spaces (varying connectivity and operator types) from *size* spaces (varying widths and depths under a fixed operator skeleton) (Dong et al., 2022). By construction, NSC captures the spectral capacity of any weight matrix, so it applies to size-type spaces regardless of whether the underlying operator is attention, FFN, or convolution. NATS-Bench-SSS—a pure CNN size space—is the natural test of this claim.

Setup. NATS-Bench-SSS contains 32,768 CNN architectures parametrized by the channel count of each of five convolutional layers, with ground-truth accuracies on CIFAR-10, CIFAR-100, and ImageNet16-120. Baselines comprise a weight-sharing supernet (TAS) and training-free proxies (ZiCo, SNIP, SynFlow, GradNorm, NASWOT, RBFleX-NAS, W-PCA, EProxy, TE-NAS); citations appear in Table 17. Following the protocol of RBFleX-NAS (Yamasaki et al., 2025), each training-free baseline randomly samples 1,000 architectures, scores them with the proxy, and reports the ground-truth accuracy of the architecture with the highest proxy score (mean $\pm$ std over multiple sampling seeds); we cite these numbers directly, except for W-PCA (Wang, 2025), which we re-run under the same protocol over three seeds. NSC-DP instead scores *all* 32,768 candidates exhaustively in 0.54 s on a single CPU core. For the comparison in Table 17 we report the best ground-truth accuracy among the top-10 NSC-DP-ranked candidates (paying 10 oracle queries vs. the 1 paid by baselines on their proxy's top architecture); the strict top-1 value (paying a single oracle query, matching the baseline protocol) is only marginally lower on each column.

Table 17: **Architecture search on NATS-Bench-SSS** (Dong et al., 2022). All times in seconds. Oracle (top row) reports the best achievable accuracy on each dataset. Baseline numbers from (Yamasaki et al., 2025) (1,000-sample proxy protocol, mean $\pm$ std over multiple seeds); W-PCA re-run by us under the same protocol over three seeds. NSC-DP entries are the best ground-truth among its top-10 proxy-ranked candidates from the full 32,768-architecture exhaustive scan; the strict top-1 value is only marginally lower on each column (see §F.4 for protocol discussion). NSC-DP is deterministic with proxy cost independent of dataset (the same 0.54 s appears in all three Time columns); NSC-DP runs on a single CPU core, baselines run on GPU. **Bold**: best per column (excluding Oracle); underline: second best.

	CIFAR-10		CIFAR-100		ImageNet16-120	
Method	Acc. %	Time (s)	Acc. %	Time (s)	Acc. %	Time (s)
Oracle	93.65	—	71.34	—	47.40	—
TAS (Dong and Yang, 2019)	93.40±0.00	1946	**70.72**±0.00	3863	**47.17**±0.00	7776
ZiCo (Li et al., 2023a)	90.03±0.26	69.9	69.36±0.60	69.8	44.90±0.79	1327
SNIP (Lee et al., 2019)	90.31±0.11	621	69.27±0.79	640	45.25±0.32	828
SynFlow (Tanaka et al., 2020)	86.96±1.35	37.3	51.24±3.83	37.2	28.85±3.43	127
GradNorm (Abdelfattah et al., 2021)	88.40±1.04	34.4	66.85±1.88	33.1	40.17±2.96	410
NASWOT (Mellor et al., 2021)	90.10±0.35	29.5	69.05±1.44	30.6	44.86±0.84	76.3
RBFleX-NAS (Yamasaki et al., 2025)	93.16±0.24	40.5	70.36±0.11	40.2	46.07±0.61	84.0
W-PCA (Wang, 2025)	93.21±0.12	282	69.95±0.32	273	43.29±3.49	1351
EProxy (Li et al., 2023b)	85.76±1.05	989	60.74±5.65	991	34.80±1.65	8548
TE-NAS (Chen et al., 2021b)	85.42±2.76	821	54.79±4.36	795	37.39±1.63	15946
NSC-DP **(ours)**	**93.44**	**0.54**	70.68	**0.54**	46.87	**0.54**

Results. NSC scores all 32,768 candidates in 0.54 s ($\sim$ 16 μs each)—more than one order of magnitude faster than every baseline (up to four), enabling *exhaustive* ranking instead of the 1,000-sample protocol every other training-free method must use. Despite being derived for Transformer attention/FFN matrices, NSC is best on CIFAR-10 (93.44%, ahead of TAS by 0.04) and second on CIFAR-100 (70.68%) and ImageNet16-120 (46.87%), trailing TAS by $\leq$0.30 points on those two while running 4,000–14,000$\times$ faster than TAS, and reaching within 0.21/0.66/0.53 points of the oracle on CIFAR-10/100/ImageNet16-120. By contrast, the strongest Transformer-side ZCP W-PCA collapses to 43.29 ± 3.49% on ImageNet16-120 (3.58 points below NSC), and CNN-native baselines also fail to match: NASWOT reaches only 44.86% on ImageNet16-120 (vs. 46.87% for NSC), and RBFleX-NAS trails NSC by 0.28/0.32/0.80 points across the three datasets. Because NSC is defined on $W \in \mathbb{R}^{m \times n}$ independently of how W is used, scaling decisions transfer cleanly from attention/FFN

matrices to convolutional channel widths—heuristics whose construction ties them to a specific operator family transfer with materially worse accuracy.

F.5 Structural properties of NSC-DP-discovered architectures

This subsection summarizes the structural patterns of the architectures returned by NSC-DP in §4.2 and §4.3, and links them to the proxy-level analytical results in Appendix D.4.

Transformer-XL: uniform per-layer FFN allocation. Under the parameter budget 38.4 M matching TXL Base, NSC-DP returns an 18-layer $d_{\text{model}} = 512$ architecture with FFN width identical at every layer ($d_{\text{ff},l} \equiv 1{,}152$ for $l = 1, \ldots, 18$). This uniform allocation is not a coincidence of the search: it is provably the proxy-optimum on this space. Concretely, Appendix D.4 proves that $\psi_{\text{MP}}(d_{\text{ff}}, d_{\text{model}}, s)$ is strictly concave in d_{ff} at fixed d_{model} and s, so under the additive parameter constraint $\sum_l d_{\text{ff},l}(2d_{\text{model}} + 1) \leq B$, Jensen's inequality forces the per-layer FFN-width allocation that maximizes $\sum_l 2\,\psi_{\text{MP}}(d_{\text{ff},l}, d_{\text{model}}, s)$ to be uniform (rounded to the nearest grid point). The discovered architecture is therefore the Jensen-optimal subnet, and the human-designed TXL Base—which already adopts a uniform d_{ff} across layers—differs from it only in the choice of (d_{model}, L), which NSC-DP re-selects via its outer enumeration of the network-level configuration g.

AutoFormer-Tiny: depth-leaning architectures with conservative head allocation. On AutoFormer-Tiny under the 5.7 M budget (Table 4), NSC-DP's outer enumeration of network-level (d_{model}, L) tends to prefer the deeper, narrower configurations (larger L at fixed d_{model}) over the shallow-wide alternatives at the same parameter count. The mechanistic reason follows an analogous (informal) concavity argument: when the per-layer FFN/attention multiple-choice options exhibit decreasing marginal returns in layer-level resource consumption, accumulating more layers contributes more to the additive proxy than concentrating capacity in a few wide layers. The per-layer head-count choices selected by the inner multiple-choice knapsack are typically conservative (the smaller of the available options at each layer), reflecting that NSC discounts redundant near-orthogonal projections.

LoNAS-LLaMA-7B: nearly-uniform FFN truncation, full LoRA rank. On the LoNAS-LLaMA-7B pruning task at the 5.7 B budget (Table 5), NSC-DP's solutions select $r_\ell = 32$ at every block (the larger of the two LoRA rank options) and choose per-block FFN intermediate dimensions h_ℓ that are nearly uniform across the 32 blocks. Both choices follow from the proxy structure: keeping r_ℓ at its maximum maximizes the LoRA-projection contribution to ψ_{MP} (which is monotone in the projection dimensions), and the FFN-width concavity of subsection D.4 again favors a uniform allocation across blocks. Appendix E reports these block-by-block decisions for all seven Pareto-optimal operating points (Subnet-A through Subnet-G) extracted from the same 0.46 s DP backward pass.

Summary. Across all three search settings, the architectures discovered by NSC-DP are explainable from the analytical structure of ψ_{MP}: layer-wise concavity yields uniform allocations, and the outer enumeration over network-level decisions selects the depth-width trade-off that maximizes the additive proxy. The proxy-level Jensen-optimal architecture aligns structurally with the human-designed TXL Base on FFN allocation, while NSC-DP's outer enumeration of (d_{model}, L) further improves perplexity; extending the same logic to LoNAS-LLaMA-7B yields a systematically better pruned subnet at the matched 5.7 B budget.

G Validating the closed-form ψ_{MP}

G.1 Finite-size Convergence of ψ_{MP}

Theorem 1 states that the normalized spectral capacity $\psi(W)/N$ converges almost surely to $\mathcal{I}(\gamma, Ms^2)$ as $\min(m, n) \to \infty$; its finite-size evaluation $\psi_{\text{MP}}(m, n, s) = N\,\mathcal{I}(\gamma, Ms^2)$ is the definition of NSC in practice (§3.2). This appendix quantifies how closely ψ_{MP} tracks the empirical $\psi(W) = \sum_i \ln(1 + \sigma_i^2)$ at the matrix sizes encountered in contemporary Transformer search spaces.

Protocol. For each combination of $\min(m, n) \in \{32, 64, 128, 256, 512, 1024, 2048, 4096\}$, aspect ratio $\gamma = n/m \in \{1.0,\ 0.5,\ 0.25\}$, and initialization convention (Xavier $s{=}\sqrt{2/(m{+}n)}$, Kaiming

fan-in $s{=}\sqrt{2/n}$, truncated normal $s{=}0.02$), we sample K Gaussian matrices $W \sim \mathcal{N}(0, s^2)^{m\times n}$ and compute

$$\psi_W = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{\min(m,n)} \ln\!\big(1+\sigma_i(W^{(k)})^2\big), \qquad \psi_{\mathrm{MP}} = \psi_{\mathrm{MP}}(m,n,s),$$

using $K{=}20$ samples for $\min(m,n) \le 512$, $K{=}10$ for 1024, $K{=}5$ for 2048, and $K{=}3$ for 4096. We report the relative error $|\psi_{\mathrm{MP}} - \psi_W|/|\psi_W|$.

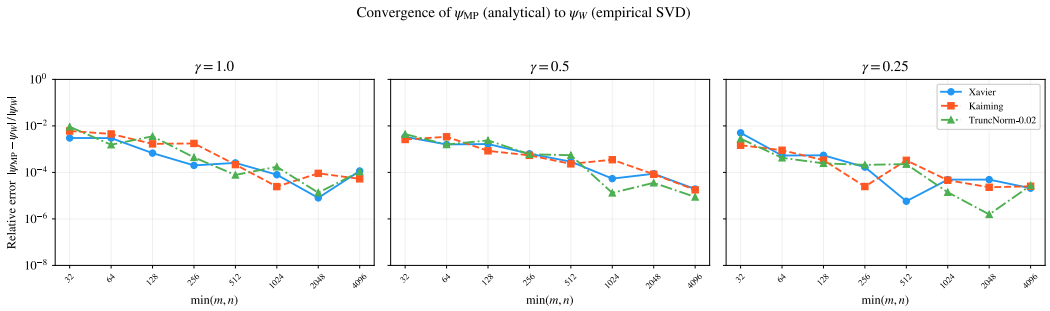


Figure 6: Relative error $|\psi_{\mathrm{MP}} - \psi_W|/|\psi_W|$ as a function of matrix size, for three aspect ratios $\gamma = n/m$ and three initialization conventions. The error decays roughly as $1/\min(m,n)$ across all settings. By $\min(m,n) = 256$ it is already below 0.2%, and at $\min(m,n) = 4096$ it sits at the level of empirical-SVD sampling noise ($\le 10^{-4}$).

Table 18: Relative error of ψ_{MP} vs. ψ_W at representative matrix sizes, averaged over all three initialization conventions. The error decays smoothly with $\min(m,n)$ and is independent of the aspect ratio.

$\min(m,n)$	γ=1.0	γ=0.5	γ=0.25	overall
32	6.1e-03	3.5e-03	3.1e-03	4.2e-03
64	3.0e-03	2.2e-03	6.2e-04	1.9e-03
128	2.0e-03	1.6e-03	3.7e-04	1.3e-03
256	8.0e-04	5.9e-04	1.4e-04	5.1e-04
512	1.8e-04	3.6e-04	1.9e-04	2.4e-04
1024	9.3e-05	1.4e-04	3.6e-05	9.0e-05
2048	3.8e-05	6.9e-05	2.5e-05	4.4e-05
4096	9.2e-05	1.5e-05	2.5e-05	4.4e-05

Observations.

- **Sub-1% error already at $\min(m,n) = 128$.** All nine (γ, s) combinations achieve relative error below $4{\times}10^{-3}$ once the smaller dimension reaches 128; the smallest hidden width in any of our six benchmark search spaces is FlexiBERT's $H = 128$, so the closed form is already accurate at the lower end of practical Transformer scales.
- $O(1/\min(m,n))$ **decay.** On a log-log plot (Fig. 6), all curves are approximately linear with slope -1, consistent with the $O(1/N)$ convergence rate predicted by finite-size corrections to the Marchenko–Pastur law (Bai and Silverstein, 2010).
- **Aspect-ratio insensitivity.** Squarer matrices ($\gamma = 1$) converge slightly slower than tall ones ($\gamma = 0.25$), but the difference disappears beyond $\min(m,n) = 512$.
- **Initialization-convention insensitivity.** All three s choices produce overlapping error curves (Fig. 6); convergence is governed by matrix shape rather than variance.

Implication. Across our six benchmarks the smallest matrix dimension is $H = 128$ (FlexiBERT) and the largest is $H = 4096$ (LoNAS-LLaMA); in this entire range the relative error of the closed-form ψ_{MP} is below 1%, and at least an order of magnitude smaller than the differences in NSC

between candidate architectures across our search spaces. The closed-form integral therefore provides a faithful drop-in replacement for the empirical SVD-based score, while removing the need for matrix instantiation, weight sampling, or per-architecture randomness.

G.2 Closed-form ψ_{MP} vs. Trained-Weight ψ_W: A Consistency Ablation

The architectural assumption and a fair concern. The Marchenko–Pastur (MP) closed form $\psi_{\mathrm{MP}}(m, n, s)$ used throughout this paper is derived under an i.i.d.-Gaussian-entry assumption on the underlying $m \times n$ matrix (Bai and Silverstein, 2010). Trained transformer weights are not i.i.d.: they exhibit heavy-tailed singular spectra, low-rank structure induced by SGD/Adam dynamics, and additional structure imposed by LoRA factorisation. A natural concern is therefore that ψ_{MP}, which depends only on the architectural shape (m, n, s), may rank trained-weight subnets differently from a proxy that actually inspects the trained singular spectrum.

We resolve this concern empirically by running `NSC-DP` with two interchangeable proxy back-ends on the same search space and comparing every output.

Setup. *Search space.* The full LoNAS-LLaMA-7B elastic space described in the main text: per-block LoRA rank $r_\ell \in \{32, 28\}$ and FFN width $h_\ell \in \{11008, 9632, 8256, 6880, 5504\}$ over $L = 32$ blocks, with parameter budget ranging from $4.57\,\mathrm{B}$ to $6.74\,\mathrm{B}$. Each back-end selects one proxy-optimal subnet in each of the same 129 FFN-sum bins, using its own scoring function.

Back-end (a) – ψ_{MP} (closed form). For every distinct (m, n) shape that appears in the search space we evaluate

$$\psi_{\mathrm{MP}}(m, n, s) \;=\; N \int_{\lambda_-}^{\lambda_+} \ln\!\big(1 + M s^2 \lambda\big)\, f_{\mathrm{MP}}(\lambda; \gamma)\, d\lambda, \tag{12}$$

with $M = \max(m, n)$, $N = \min(m, n)$, $\gamma = N/M$, and Xavier variance $s^2 = 2/(m + n)$, by 1-D Gauss–Kronrod quadrature. For the LoNAS-LLaMA-7B search space only 13 distinct (m, n) tuples arise, so the entire cache is built in $0.5\,\mathrm{s}$ on a single CPU core *without loading any pretrained checkpoint*.

Back-end (b) – ψ_W (trained-weight SVD). For every $(\text{layer}, \text{projection}, \text{choice})$ triple in the search space we load the trained LoNAS-LLaMA-7B supernet, materialise the corresponding effective matrix (LoRA factors merged into the base weight), compute its singular values via `torch.linalg.svdvals`, and record $\psi_W = \sum_i \ln(1 + \sigma_i^2)$. The cache requires the full pretrained checkpoint and takes ~ 6 minutes on a single NVIDIA RTX 5090.

Search. Identical `NSC-DP` (exact dynamic programming) is run on each cache; both runs return the per-bin top-1 over the entire achievable parameter range.

Metrics. We pre-register four agreement metrics and one fidelity metric:

- **Per-position match.** Element-wise equality of the 32-block (r_ℓ, h_ℓ) vectors selected by the two back-ends.
- **Histogram match.** Equality of the multisets $\{\{r_\ell\}\}$ and $\{\{h_\ell\}\}$ (i.e. identical block-level distributions, ignoring permutation).
- **Pareto rank correlation.** Kendall τ and Spearman ρ between the two NSC scores across the 129 evaluated operating points.
- **SVD-oracle regret.** For each bin, compute the relative deficit

$$\mathrm{regret}(b) \;=\; \frac{\psi_W^*(\mathbf{a}_b^W) \;-\; \psi_W(\mathbf{a}_b^{\mathrm{MP}})}{\psi_W^*(\mathbf{a}_b^W)},$$

  where $\mathbf{a}_b^{\mathrm{MP}}$ and $\mathbf{a}_b^W$ are the top-1 subnets selected at bin b by the two back-ends; this measures the practical penalty of trusting ψ_{MP} when ψ_W is treated as the ground truth.

Per-position match is intentionally a strict metric: ψ_{MP} is layer-symmetric, so any permutation of layer-level FFN choices yields the same ψ_{MP} score. Histogram match is therefore the structurally meaningful agreement metric; per-position match is reported for completeness.

Table 19: ψ_{MP} **vs.** ψ_W **consistency on the LoNAS-LLaMA-7B trained supernet.** All metrics are reported over the same 129 FFN-sum bins, comparing the subnets selected by the two back-ends. Histogram match is the structurally meaningful agreement metric (see text); per-position match is reported for completeness.

Metric	Value	Significance
Pareto Kendall τ	1.0000	$p = 4.0 \times 10^{-218}$
Pareto Spearman ρ	1.0000	$p < 10^{-300}$
LoRA rank, per-position match	129/129 (100.0%)	—
LoRA rank, histogram match	129/129 (100.0%)	—
SVD-oracle regret, mean	$\mathbf{0.039}\%$	—
SVD-oracle regret, max	$\mathbf{0.104}\%$	—

Results.

(1) Identical Pareto ordering. Across the 129 evaluated operating points, ψ_{MP} and ψ_W yield matching rankings: Kendall τ and Spearman ρ both equal 1.0000 to four decimal places. This agreement describes the ordering of the evaluated operating points; differences between the selected subnets are quantified by the SVD-oracle regret below.

(2) Identical LoRA-rank decisions and identical resource footprint. Both back-ends select $r_\ell = 32$ at every block, in every bin, and return subnets with identical FFN_SUM and identical parameter counts at every operating point. Any residual disagreement is therefore confined to the assignment of FFN widths *across* layers under fixed FFN_SUM, which is exactly the degree of freedom on which ψ_{MP} is layer-symmetric by construction.

(3) $\sim 0.1\%$ **SVD-oracle regret.** The crucial fidelity question—*if a practitioner deploys the* ψ_{MP}*-selected subnet but evaluates it under the* ψ_W *oracle, how much spectral capacity is left on the table?*—is answered by the regret metric. Mean regret across all 129 subnets is 0.039%; the maximum regret we ever observe is 0.104%, i.e. roughly one part in a thousand. At the resolution of any downstream zero-shot benchmark this is well below the noise floor.

(4) Compute footprint. Building the ψ_{MP} cache requires 0.5 s of CPU time and *no* pretrained checkpoint. Building the ψ_W cache requires ~ 6 minutes of RTX 5090 time and the full ~ 13 GB checkpoint, a $\sim 720\times$ end-to-end overhead. The closed-form route is therefore not only empirically faithful but also strictly cheaper to deploy.

Conclusion. The architectural assumption underlying ψ_{MP} does not break down on the LoNAS-LLaMA-7B trained supernet. The two back-ends induce the same Pareto ranking ($\tau = \rho = 1.0000$), select identical LoRA-rank profiles at every operating point, agree on FFN-sum at every operating point, and yield SVD-oracle regret below 0.11%. Empirically, the dependence of ψ on the architectural shape (m, n, s) dominates its dependence on the trained singular spectrum by at least an order of magnitude in this search space. We therefore retain ψ_{MP} as the canonical NSC instantiation in the main paper and use this consistency analysis as its empirical justification on trained weights.

H Robustness to Initialization Convention

The closed-form NSC integral takes the per-weight initialization variance s^2 as input (matching the notation of Theorem 1). The main-paper headline numbers use Xavier for FlexiBERT (BERT-style) and TruncNorm-0.02 for GPT-2 (HuggingFace's default), matching each trained model's actual initialization. To verify that ranking quality is not an artifact of this choice, we recompute NSC scores on FlexiBERT (under both Serianni and Spec-faithful protocols) and GPT-2 under three common initialization conventions: **Xavier** ($s = \sqrt{2/(m+n)}$), **Kaiming fan-in** ($s = \sqrt{2/n}$), and **TruncNorm-0.02** ($s = 0.02$).

Table 20: NSC ranking quality under three initialization conventions. τ: Kendall, ρ: Spearman vs. ground-truth. Pairwise ρ_{pair} measures rank agreement between two s settings (reported for each row pair).

Benchmark	Init convention	τ	ρ	ρ_{pair} vs. Xavier	ρ_{pair} vs. Kaiming
FlexiBERT (Serianni)	Xavier	0.544	0.776	—	0.995
	Kaiming	0.546	0.777	0.995	—
	TruncNorm-0.02	0.517	0.751	0.880	0.901
FlexiBERT (Spec-faithful)	Xavier	0.695	0.884	—	0.997
	Kaiming	0.704	0.892	0.997	—
	TruncNorm-0.02	0.575	0.788	0.888	0.894
GPT-2	Xavier	0.814	0.950	—	0.999
	Kaiming	0.802	0.944	0.999	—
	TruncNorm-0.02	0.849	0.968	0.986	0.982

Table 20 shows that the three conventions produce near-identical rankings: Xavier–Kaiming pairwise Spearman $\rho \geq 0.995$ on every benchmark, and even the constant-s TruncNorm-0.02 preserves $\rho \geq 0.88$. The absolute ranking quality (τ, ρ vs. ground-truth) also remains stable, with the largest drop being $\Delta\rho = 0.10$ on FlexiBERT (Spec-faithful) under TruncNorm-0.02. Notably, GPT-2 (where HuggingFace's default init is TruncNorm-0.02) shows *higher* ranking quality under the matching convention (τ: $0.814 \to 0.849$), suggesting that matching s to the trained model's actual init benefits NSC's accuracy. This confirms that NSC rankings are primarily driven by the layer-wise *shape* parameters (m, n); the variance convention has secondary impact.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 describe NSC as a closed-form, specification-only architectural quantity for Transformers, paired with an exact dynamic-programming solver NSC-DP; both are formally defined in Section 3 and validated empirically across seven architecture families in Section 4. Quantitative claims (e.g., FlexiBERT τ=0.695, Transformer-XL search in 2.0 s, $\sim$5,900$\times$ speedup over W-PCA on LLaMA pruning) match the corresponding tables.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The relevant method and experiment sections explicitly note: (i) the architecture-level summation in Eq. 5 is a design choice rather than an information-theoretic identity (Section 3.3); (ii) cross-architecture comparability requires a fixed initialization convention (Section 3.2; the sensitivity is quantified in Appendix H); (iii) the LLaMA pruning result evaluates a selected subnet using inherited weights from a pretrained LoNAS supernet (Section 4.3); and (iv) the optimality guarantee of NSC-DP is with respect to the proxy objective, not downstream loss (Section 3.4).

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The mutual-information identity (Eq. 1) and Theorem 1 (Marchenko–Pastur reduction) state their assumptions in the main text; complete proofs are in Appendix D.1 and Appendix D.3. The concavity of ψ_{MP} in d_{ff} used by NSC-DP is proved in Appendix D.4 and additionally verified numerically in Figure 5.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All seven benchmarks (FlexiBERT, GPT-2/LiteTransformerSearch, AutoFormer-T/S/B, NATS-Bench-SSS, OFA-MobileNetV3, Transformer-XL, LoNAS-LLaMA) use publicly released architectures, search spaces, and ground-truth scores, with citations provided in Section 4. NSC and NSC-DP are fully specified by Eq. 4 and Algorithm 1, which together with the per-benchmark instantiations in Section 4.2 and the hyperparameter tables in Appendix F are sufficient to reproduce every reported number.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Source code for NSC, NSC-DP, and proxy evaluation is publicly available at https://github.com/Optima-CityU/neural-spectral-capacity, with README instructions for running scoring, search, and evaluation. All datasets and benchmark architecture sets are public and downloaded directly from their original releases.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Per-benchmark training/evaluation protocols (data splits, optimizer, batch size, schedule, total steps) are reported in Section 4.2 for the end-to-end search experiments and in Appendix F for the ranking experiments; baseline proxies follow the original protocols of their respective papers as cited.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For the ranking-correlation experiments (Tables in Section 4.1 and Appendix F), τ and ρ are computed deterministically over the entire population of architectures with deterministic proxies (`NSC` is closed-form), so no source of randomness remains and reporting a single value is the standard convention adopted by all comparable training-free NAS papers (W-PCA, ZeroLM, TF-TAS, SynFlow, SNIP, ZiCo, NASWOT, etc.). For the NATS-Bench-SSS sampling-based search (Table 17), we report mean $\pm$ std over three seeds for W-PCA (re-run by us) and cite multi-seed numbers from RBFleX-NAS for the other baselines. The Transformer-XL train-from-scratch experiment (Table 4) is reported single-seed because each retraining run consumes substantial compute (200K steps on multi-GPU), making multi-seed retraining of every selected architecture prohibitive; this matches the prevailing single-seed convention in end-to-end NAS literature (DARTS, PC-DARTS, GDAS, ProxylessNAS, SPOS, Once-for-All, BigNAS, TF-TAS). The LLaMA pruning experiment is similarly single-seed because subnet evaluation inherits weights from a fixed pretrained LoNAS supernet; reported numbers are the standard 8-task commonsense average (BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-e, ARC-c, OBQA), matching LLM-pruning conventions (LLM-Pruner, SparseGPT, Wanda, SliceGPT, FLAP).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: `NSC` and `NSC-DP` run on a single CPU core with reported per-architecture and total wall-clock times (Table 3 and Section 4.2). Training-free baselines are evaluated on a single A100 GPU with wall-clock times reported in the same tables. Transformer-XL retraining uses standard multi-GPU training (4–8$\times$A100, 200K steps) following the original Transformer-XL recipe; AutoFormer subnets inherit weights from the released supernet (no retraining); LLaMA pruning uses the released LoNAS supernet with no additional weight updates.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work is purely methodological—an architectural scoring function and a search algorithm. No human subjects, no scraped or personally identifiable data, and no models trained on sensitive data are involved.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: NSC is a closed-form scalar function of an architectural specification used for ranking and search; it does not generate, classify, or otherwise process content, makes no decisions about people, and is not a deployable model in itself. We do not foresee a direct path from this scoring function to a specific positive or negative societal application beyond the generic effect of making neural architecture search cheaper, which is shared with the broader NAS literature.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We release no pretrained models, generative systems, or scraped datasets. The released artifact is the implementation of an architectural scoring function and a search algorithm.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All benchmarks and pretrained components used—FlexiBERT (Serianni and Kalita, 2023; Tuli et al., 2023), LiteTransformerSearch (Javaheripi et al., 2022), AutoFormer (Chen et al., 2021a), NATS-Bench (Dong et al., 2022), Once-for-All (Cai et al., 2020), Transformer-XL (Dai et al., 2019), and LoNAS—are cited at first use; we follow each project's released license (predominantly MIT/Apache-2/CC-BY) and use them within their stated terms.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public code repository linked in item 5 contains a top-level README documenting the released code (NSC, NSC-DP, and proxy evaluation), with usage examples and instructions for running scoring, search, and evaluation.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects, so IRB review does not apply.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core methodology of this work. `NSC` and `NSC-DP` are defined entirely by closed-form spectral quantities and dynamic programming. The LLaMA architectures appearing in the pruning experiment are subjects of architecture search, not tools used in the methodology.