WHEN IS A REPRESENTATION SUFFICIENT? PROBE $\mathcal{V}$-INFORMATION FOR NEURAL ARCHITECTURE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

When is a neural representation sufficient for a task? A representation Z is sufficient if $I(X;Y \mid Z) = 0$, meaning no task-relevant information remains in the input. This provides a natural criterion for deciding when additional model capacity is unnecessary. However, estimating such information-theoretic quantities in high-dimensional neural networks is intractable in practice. We introduce $\mathcal{V}$-NAS, a progressive architecture search method that replaces intractable Shannon information with probe-based $\mathcal{V}$-information, a directly computable measure of *extractable predictive information*. Using the cross-entropy of a restricted probe, our method determines whether increasing width or depth yields meaningful gains, enabling principled capacity control. Across multiple datasets and architecture families, $\mathcal{V}$-NAS discovers compact architectures that match the performance of significantly larger baselines at substantially lower cost. These results demonstrate that representation sufficiency can serve as a practical and interpretable foundation for neural architecture design.

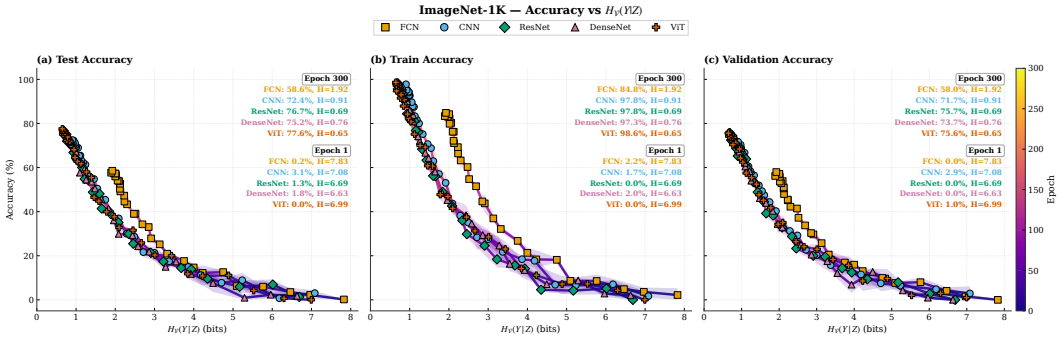

Figure 1: **ImageNet-1K: Accuracy vs. $\mathcal{V}$-entropy $H_{\mathcal{V}}(Y \mid Z)$ across five architecture families.** Each curve traces a single architecture over 300 training epochs (shaded bands show 90% CIs). **(a) Test accuracy** is tightly anti-correlated with $H_{\mathcal{V}}(Y \mid Z)$ across all families, confirming that probe loss is an architecture-agnostic predictor of generalization. **(b) Train accuracy** follows a similar trajectory with tighter CIs, showing that the $\mathcal{V}$-information signal tracks optimization progress. **(c) Validation accuracy** closely mirrors test accuracy, indicating that the probe-loss criterion does not overfit the validation split and is safe to use for architecture decisions during the search.

1 INTRODUCTION

Neural architecture search (NAS) aims to automate the design of neural networks by selecting architectures that achieve strong predictive performance while remaining computationally efficient. Since the earliest reinforcement-learning and evolutionary approaches (Zoph & Le, 2017; Zoph et al., 2018; Real et al., 2019), a large body of work has sought to reduce the cost of search through progressive architecture growth via sequential model-based optimization (Liu et al., 2018), multi-fidelity optimization (Li et al., 2018), differentiable relaxation (Liu et al., 2019; Xie et al., 2019), and training-free proxies (Mellor et al., 2021; Chen et al., 2021a; Abdelfattah et al., 2021). These

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

methods share a common practical objective: to identify an architecture with an effective trade-off between accuracy and cost.

A central difficulty in this setting is that architecture quality must typically be inferred from imperfect surrogate signals, such as validation accuracy and training loss. Validation accuracy obtained early in training is noisy and may favor architectures that optimize quickly rather than generalize effectively (Li et al., 2018; Domhan et al., 2015). Training loss provides limited information, as it primarily reflects optimization progress rather than final generalization or computational cost. In contrast to the above two, zero-cost and low-cost proxies provide efficient ranking criteria (Mellor et al., 2021; Chen et al., 2021a), but these criteria are generally heuristic and architecture-dependent. None of these signals directly answers a more fundamental question: *does the current representation already contain all task-relevant information, or is additional capacity still warranted?*

An information-theoretic criterion and its practical barrier. From a representation perspective, this problem can be framed in information-theoretic terms. Let Z denote the representation produced by a network for predicting a target Y from input X. Conditional Mutual Information (CMI) yields:

$$I(X;Y \mid Z) = H(Y \mid Z) - H(Y \mid X), \tag{1}$$

where $H(Y \mid X, Z) = H(Y \mid X)$ since Z is a deterministic function of X, quantifies the residual task-relevant information in X that is not captured by Z. A representation is *sufficient* when $I(X;Y \mid Z) = 0$, in which case no additional capacity is required to improve prediction. This suggests a natural growth rule: expand the network until $I(X;Y \mid Z)$ ceases to decrease, but the criterion is impractical as stated. Binning-based estimators are biased and do not scale well to high dimensions (Goldfeld et al., 2019), while k-nearest-neighbor methods such as KSG (Kraskov et al., 2004) suffer from the curse of dimensionality. Extensions for CMI (Frenzel & Pompe, 2007) and matrix-based Rényi entropy estimators (Giraldo et al., 2015; Jizba & Arimitsu, 2004) alleviate some issues but introduce additional assumptions or measure different entropy notions. In short, building architecture decisions on unreliable estimates is fragile.

From estimation to computation. To address this limitation, we replace Shannon quantities with operationally defined ones. Following the $\mathcal{V}$-information framework (Xu et al., 2020a), we measure predictive uncertainty through the cross-entropy of a restricted probe:

$$H_{\mathcal{V}}(Y \mid Z) = \inf_{f \in \mathcal{V}} \mathbb{E}\big[-\log f(Y \mid Z)\big]. \tag{2}$$

This quantity is directly computable via empirical risk minimization, and it measures *extractable* predictive information under a fixed hypothesis class $\mathcal{V}$ rather than the total information in the Shannon sense. Crucially, the probe loss is not a noisy *approximation* to $H(Y \mid Z)$; it is the quantity of interest.

Progressive capacity expansion. Building on this perspective, we formulate architecture growth as a progressive capacity expansion problem. At each stage, we evaluate whether increasing width or depth produces a meaningful 2 in probe-based predictive uncertainty, standardized by its variability across samples and adjusted for the complexity cost of expansion. Both width and depth decisions are governed by a single information-theoretic criterion: accept the expansion only when the signal-to-noise ratio of the per-sample probe-loss improvement exceeds a cost-adjusted threshold. Empirically, we show that early probe measurements correlate with final model quality and that the resulting architectures are competitive with standard hand-designed baselines of comparable complexity under matched training protocols. Contributions of this work can be summarized as follow:

1. We formulate neural architecture growth as a **representation sufficiency problem** based on conditional Mutual Information (MI).
2. We introduce a **computable surrogate** using probe-based $\mathcal{V}$-information, avoiding intractable Shannon estimation.
3. We propose a **cost-aware expansion rule** that unifies width and depth decisions through a single information-theoretic criterion.

Related work. Due to space constraints, we defer review of prior work to Appendix C.

2 REPRESENTATION SUFFICIENCY VIA INFORMATION THEORY

We consider a supervised setting with input–label pairs (X, Y) and a deterministic network that forms a Markov chain:

$$Y \text{ — } X \to Z_1 \to \cdots \to Z_K \to \hat{Y}, \tag{3}$$

where $Z_i = f_i \circ \cdots \circ f_1(X)$ for $1 \le i \le K$ are hidden-layer representations, and $\hat{Y}$ is the predicted output. Our goal is a principled criterion for expanding network capacity (width and depth) during architecture search. We begin with the Shannon-theoretic ideal, identify why it is impractical to estimate, and resolve the gap with $\mathcal{V}$-information.

2.1 SHANNON-THEORETIC IDEAL

Theorem 2.1 (Residual Information and Sufficiency)**.** *Let $Z_i = f_i \circ \cdots \circ f_1(X)$ be the representation at layer i. Then:*

1. *The CMI decomposes as:*

$$I(X; Y \mid Z_i) = H(Y \mid Z_i) - H(Y \mid X), \tag{4}$$

   where $H(Y \mid X, Z_i) = H(Y \mid X)$ since $Z_i = f(X)$ is deterministic.

2. *Z_i is a sufficient statistic for Y with respect to X, i.e., $p_{Y|Z_i}(y \mid z_i) = p_{Y|X}(y \mid x)$ for p-a.e. (x, z_i) in the support, if and only if $I(X; Y \mid Z_i) = 0$.*

Theorem 2.2 (Representation Complexity Decomposition)**.** *For continuous representations $Z_i \in \mathbb{R}^{m_i}$ whose density p_{Z_i} exists with respect to Lebesgue measure, define the* density-ratio functional*:*

$$\mathcal{J}(X, Z_i, Y) \;=\; \mathbb{E}\left[\log \frac{p(X, Z_i, Y)}{p(X, Z_i)\, p(Z_i, Y)}\right], \tag{5}$$

where the expectation is taken over $(X, Z_i, Y) \sim p_{X,Z_i,Y}$ with the integral over $\mathcal{X} \times \mathcal{Z}_i$ and the sum over $\mathcal{Y}$. Define the differential entropy *$h(Z_i) = -\int p_{Z_i}(z_i) \log p_{Z_i}(z_i)\, dz_i$. Then the following identity holds:*

$$\mathcal{J}(X, Z_i, Y) \;=\; h(Z_i) \;+\; I(X; Y \mid Z_i). \tag{6}$$

When Z_i is a sufficient statistic ($I(X; Y \mid Z_i) = 0$, Theorem 2.1), this reduces to:

$$\mathcal{J}(X, Z_i, Y) = h(Z_i), \tag{7}$$

so that minimizing representation complexity $\min h(Z_i)$ subject to sufficiency is equivalent to minimizing $\mathcal{J}$.

Remark 2.3 (Scope of the Shannon formulation). Two pathologies limit the practical scope of Theorem 2.2. **(i) Singular joint distribution.** Although the density-ratio expectation in Eq. equation 5 is finite (the singularities from the shared variable Z_i cancel in the ratio; see Appendix D), the quantity $\mathcal{J}$ does *not* equal the Shannon mutual information $I\big((X, Z_i);\, (Z_i, Y)\big)$, which is formally infinite because the joint distribution of (X, Z_i) and (Z_i, Y) is singular with respect to the product of its marginals. **(ii) Manifold support.** When $\dim(Z_i) > \dim(X)$, common in early, width-expanding layers, the deterministic map $Z_i = f(X)$ concentrates the distribution of Z_i on a lower-dimensional manifold, so p_{Z_i} does not exist with respect to Lebesgue measure and $h(Z_i)$ is undefined (or $-\infty$). Both pathologies are absent from the $\mathcal{V}$-information framework (§2.3), which replaces differential entropy with empirical probe loss and is always finite.

Theorem 2.1 characterizes the ideal stopping condition: the representation is sufficient when the residual $I(X; Y \mid Z_i) = 0$. In principle, one could grow the network until this residual is negligible. However, for a deterministic network the Data Processing Inequality (DPI) implies $H(Y \mid Z_{i+1}) \ge H(Y \mid Z_i)$ whenever $Z_{i+1} = f_{i+1}(Z_i)$, so the Shannon residual $I(X; Y \mid Z_i) = H(Y \mid Z_i) - H(Y \mid X)$ is monotonically *non-decreasing* with depth. This means Shannon MI is inherently insensitive to depth-dependent improvements in representation quality, and it cannot serve as a useful signal for when to stop adding layers. Furthermore, evaluating the criterion requires estimating two conditional entropies, $H(Y \mid Z_i)$ and $H(Y \mid X)$, from finite samples of high-dimensional representations, a problem that is practically intractable with standard tools. These limitations motivate a shift from estimating Shannon information to a framework in which the probe loss *is* the quantity of interest.

2.2 THE ESTIMATION GAP

In practice, the Shannon quantities in Eq. equation 4 are approximated using cross-entropy losses of trained predictors (probes). Specifically, conditional entropies are upper bounded as:

$$\mathcal{L}_{\mathrm{CE}}^{\mathrm{probe}}(Z_i) \geq H(Y \mid Z_i), \qquad \mathcal{L}_{\mathrm{CE}}^{\mathrm{ref}}(X) \geq H(Y \mid X). \tag{8}$$

Their difference $\widehat{\mathrm{CMI}} = \mathcal{L}_{\mathrm{CE}}^{\mathrm{probe}}(Z_i) - \mathcal{L}_{\mathrm{CE}}^{\mathrm{ref}}(X)$ is the difference of two upper bounds, which does *not* bound the true $I(X; Y \mid Z_i)$ from either direction. A probe with limited capacity on raw inputs X may achieve a loose bound, while the same probe on learned features Z_i achieves a tight one, yielding $\widehat{\mathrm{CMI}} \approx 0$ even when significant residual information remains.

Relationship to the experimental ablation. The "Shannon CMI" ablation in our experiments uses a *different* formula from the gap described above: it computes the adjacent-layer probe-loss difference $\mathcal{L}_{\mathrm{CE}}^{\mathcal{V}}(Z_{i-1}, Y) - \mathcal{L}_{\mathrm{CE}}^{\mathcal{V}}(Z_i, Y)$, which is the *unstandardized* numerator $\bar{\delta}$ of our SNR criterion (Eq. 14). This quantity is mathematically a $\mathcal{V}$-entropy difference, not a Shannon CMI estimate. Its instability in the ablation therefore stems from the absence of per-sample standardization (division by s_δ) and cost-adjusted thresholding ($\gamma \cdot (1 + \log_2 r)$), not from the intractability of Shannon estimation. We retain the label "Shannon CMI" in the ablation to reflect its naïve, unstandardized use of the probe-loss signal, but emphasize that the mathematical distinction from our $\mathcal{V}$-NAS criterion is standardization and cost adjustment, not the entropy notion itself.

2.3 BRIDGING THE THEORY AND REALITY GAP WITH $\mathcal{V}$-INFORMATION

We adopt the framework of $\mathcal{V}$-*information* (Xu et al., 2020a), which replaces intractable Shannon quantities with operationally defined, directly computable alternatives. The key insight of $\mathcal{V}$-information is to abandon the goal of estimating Shannon entropy altogether: rather than treating probe loss as a noisy *approximation* to $H(Y \mid Z)$, it treats the minimum achievable probe loss *itself* as the fundamental quantity.

Let $\mathcal{V}$ denote a family of predictors $f : \mathcal{X} \cup \{\emptyset\} \to \mathcal{P}(\mathcal{Y})$, where $\mathcal{P}(\mathcal{Y})$ is the set of probability distributions over $\mathcal{Y}$, and $\emptyset$ represents the absence of input.

Definition 2.1 ($\mathcal{V}$-Entropy)**.** Let $(X, Y) \sim p_{X,Y}$. The *predictive conditional* $\mathcal{V}$*-entropy* of Y given X is:

$$H_{\mathcal{V}}(Y \mid X) = \inf_{f \in \mathcal{V}} \mathbb{E}_{(x,y)\sim p_{X,Y}}\big[-\log f(x)(y)\big], \tag{9}$$

where $f(x)(y)$ denotes the probability assigned to label y by predictor f given input x. The *marginal* $\mathcal{V}$*-entropy* (the optimal risk without side information) is:

$$H_{\mathcal{V}}(Y) = H_{\mathcal{V}}(Y \mid \emptyset) = \inf_{f \in \mathcal{V}} \mathbb{E}_{y\sim p_Y}\big[-\log f(\emptyset)(y)\big]. \tag{10}$$

Definition 2.2 ($\mathcal{V}$-information)**.** The $\mathcal{V}$*-information* from X to Y is the reduction in predictive risk gained by observing X:

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y) - H_{\mathcal{V}}(Y \mid X). \tag{11}$$

Unlike Shannon MI, $I_{\mathcal{V}}$ is (i) directly computable via empirical risk minimization, (ii) explicitly dependent on the hypothesis class, and (iii) *not* subject to the Data Processing Inequality. For a restricted $\mathcal{V}$, a well-trained representation Z_i can satisfy $I_{\mathcal{V}}(Z_i \to Y) \geq I_{\mathcal{V}}(X \to Y)$; this reversal is precisely the purpose of representation learning, and it makes the probe signal *sensitive to representation quality* in a way that Shannon MI is not. Thus, $I_{\mathcal{V}}$ serves as a computable surrogate for the intractable Shannon objective in Eq. 4, aligning the theory with what probes actually measures.

Definition 2.3 (Usable Information Gain)**.** The usable information gain at layer i is the incremental reduction in probe loss:

$$\Delta_{\mathcal{V}}(Z_i) = \mathcal{L}_{\mathrm{CE}}^{\mathcal{V}}(Z_{i-1}, Y) - \mathcal{L}_{\mathrm{CE}}^{\mathcal{V}}(Z_i, Y). \tag{12}$$

The total improvement telescopes: $H_{\mathcal{V}}(Y \mid X) - H_{\mathcal{V}}(Y \mid Z_K) = \sum_{i=1}^{K} \Delta_{\mathcal{V}}(Z_i)$.

This layerwise decomposition provides the building block for our architecture search: each capacity expansion (width or depth) is accepted only when the resulting usable information gain exceeds a cost-adjusted threshold.

3 $\mathcal{V}$-NAS: SUFFICIENCY VIA $\mathcal{V}$-INFORMATION FOR NEURAL ARCHITECTURE SEARCH

We translate the $\mathcal{V}$-information framework into a concrete search algorithm. The key design principle is that width expansion and depth extension are both capacity increases governed by the same criterion: per-sample usable information gain, standardized by its variability and adjusted for the cost of expansion.

Let $\mathcal{A}$ and $\mathcal{B}$ denote two configurations producing representations $Z^{\mathcal{A}}$ and $Z^{\mathcal{B}}$, with linear probes $f^{\mathcal{A}}, f^{\mathcal{B}} \in \mathcal{V}$ trained under identical settings. For each sample $(x_n, y_n) \in \mathcal{D}_{\text{probe}}$, the paired difference in per-sample $\mathcal{V}$-entropy is:

$$\delta_n = \underbrace{\big(-\log f^{\mathcal{A}}(z_n^{\mathcal{A}})(y_n)\big)}_{\ell_n^{\mathcal{A}}} - \underbrace{\big(-\log f^{\mathcal{B}}(z_n^{\mathcal{B}})(y_n)\big)}_{\ell_n^{\mathcal{B}}}, \tag{13}$$

where $\delta_n > 0$ indicates that $\mathcal{B}$ assigns higher probability to the correct label. The *standardized information gain* is:

$$\mathrm{SNR}(\mathcal{A}, \mathcal{B}) = \frac{\bar{\delta}}{s_\delta + \epsilon}, \qquad \bar{\delta} = \tfrac{1}{N}\textstyle\sum_n \delta_n, \qquad s_\delta = \sqrt{\tfrac{1}{N-1}\textstyle\sum_n (\delta_n - \bar{\delta})^2}. \tag{14}$$

The numerator $\bar{\delta}$ estimates the usable information gain $\Delta_{\mathcal{V}}$ on held-out data; the denominator measures consistency across the data distribution. Here $\epsilon > 0$ is a small numerical constant used for stability (e.g., $\epsilon = 10^{-6}$). This standardization ensures that improvements are not only large in expectation but also consistent across samples.

Definition 3.1 (Capacity Expansion Criterion)**.** Configuration $\mathcal{B}$ is accepted over $\mathcal{A}$ iff:

$$\mathrm{SNR}(\mathcal{A}, \mathcal{B}) > \gamma \cdot \big(1 + \log_2 r\big), \qquad r = \mathcal{C}_i(\mathcal{B})/\mathcal{C}_i(\mathcal{A}), \tag{15}$$

where $\mathcal{C}_i(\cdot)$ denotes the parameter count of the *active stage i only* (the frozen prefix stages $1, \ldots, i{-}1$ are excluded, since their cost is already committed and should not penalize further expansion). $\gamma = 0.2$ (corresponding to Cohen's "small effect" threshold) is fixed across all experiments.

The search builds the network stage by stage, alternating with width selection and depth validation. Data is partitioned into four non-overlapping splits: $\mathcal{D}_{\text{train}}$ (network and probe training), $\mathcal{D}_{\text{probe}}$ (probe evaluation and architecture decisions), $\mathcal{D}_{\text{val}}$ (model selection across seeds), and $\mathcal{D}_{\text{test}}$ (final metrics, locked until evaluation). Exact split sizes and non-overlap assertions are in Appendix H.2.

Width selection. At each stage i, width is determined by ascending geometric expansion from $w_{\text{start}} = w_{i-1}$ (or $w_{\min}$ for stage 1). Two copies of the network are built, stages $1, \ldots, i{-}1$ frozen, stage i at widths w and $2w$ with fresh He initialization (He et al., 2015), and trained for E_{search} epochs. Linear probes are trained on $\mathcal{D}_{\text{train}}$ features and evaluated on $\mathcal{D}_{\text{probe}}$. If Eq. equation 15 accepts the doubling, set $w \leftarrow 2w$ and repeat; otherwise, lock $w_i = w$.
Depth selection. After locking w_i, we compare the cached model/probe from stage $i{-}1$ against the newly locked stage i via the same SNR criterion. If the depth check fails, the search halts and outputs the architecture through stages $1, \ldots, i{-}1$. Width and depth decisions use identical machinery; the only distinction is the tested capacity axis. Schematic flowchart of the method summarized in Fig. 2

4 EXPERIMENTS

Methods. Details of baseline methods has been provided in Section H.1 in Appendix.
Datasets. We evaluate on four benchmark datasets: CIFAR-10/100 (Krizhevsky & Hinton, 2009), Tiny-ImageNet (Le & Yang, 2015), and ImageNet-1K (Deng et al., 2009). Full preprocessing details (resolution, normalization, augmentation) are in Appendix H.2.
Architecture families. We instantiate $\mathcal{V}$-NAS for five backbone families, each representing a distinct inductive bias: **(i). FCN:** fully connected layers (no spatial structure); **(ii). CNN:** plain Conv-BN-GELU blocks following VGG (Simonyan & Zisserman, 2015); **(iii). ResNet:** residual BasicBlocks with skip connections (He et al., 2016); **(iv). DenseNet:** dense blocks with feature concatenation (Huang et al., 2017); **(v). ViT:** vision transformer encoder blocks with multi-head self-attention (Dosovitskiy et al., 2021), using the DeiT training recipe modifications (Touvron et al.,

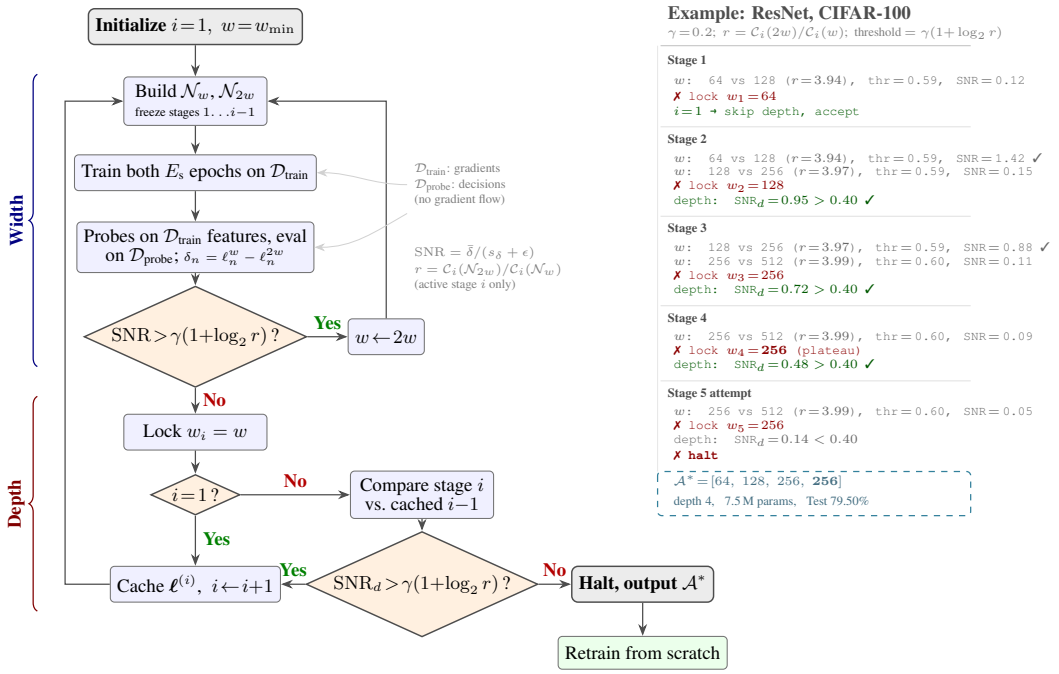

Figure 2: $\mathcal{V}$**-NAS search loop.** *Left:* **Width** (inner loop): w vs. $2w$ via paired probe losses; accept if cost-adjusted SNR $> \gamma$. **Depth** (outer loop): stage i vs. cached $i-1$; rejection halts the search. Both use the same criterion (Definition 3.1) with $r = \mathcal{C}_i(\mathcal{B})/\mathcal{C}_i(\mathcal{A})$ computed over *active-stage parameters only*. *Right:* ResNet trace on CIFAR-100 ($\gamma = 0.2$). BasicBlock width doubling gives $r \approx 4$ (quadratic in width), threshold ≈ 0.60. The search discovers widths $[64, 128, 256, \mathbf{256}]$ at depth 4; the bold plateau marks a rejected doubling, and the stage-5 depth check terminates.

2021) (but *without* distillation or MixUp/CutMix; see caption of Table 1). The *same algorithm* with the *same* $\gamma = 0.2$ discovers depth and per-stage width for every family. Discovered architectures are retrained from scratch under an identical protocol (optimizer, augmentation, epoch budget) as hand-designed baselines (ResNet-18/50/101, DenseNet-121, ViT-B/16), ensuring that accuracy differences reflect architecture choices alone. Block definitions and stage-net abstractions are in Appendix H.3.

Optimization. We use Muon (Jordan et al., 2024) for 2D weight matrices and AdamW (Loshchilov & Hutter, 2019) for 1D parameters, biases, and classifier heads, following Bernstein (2025). Linear probes are trained with Adam (lr $= 10^{-3}$, 50 epochs) on frozen representations.

Evaluation protocol. Evaluation protocol. All results are reported as mean $\pm$ standard deviation. Search cost is reported as wall-clock GPU-hours on 2$\times$ RTX A6000 (48 GB). For prior NAS methods we do not re-run the search phase; we take the architecture reported in each original paper and retrain it from scratch under our protocol, so all accuracy, parameter and $H_{\mathcal{V}}$ entries for those methods are our own measurements. *Only* the search costs are quoted from the original publications; these were obtained on different hardware and should be interpreted as indicative compute budgets rather than hardware-normalized comparisons (Table 1).

4.1 Results & Discussion

Probe loss predicts downstream performance. Figure 1 establishes the empirical foundation for the method: across all five architecture families, test, train, and validation accuracy are tightly anti-correlated with $H_{\mathcal{V}}(Y \mid Z)$ throughout training. The trajectories reveal two regimes, an early information-acquisition phase (epochs 1–50, steep probe-loss descent) and a late saturation phase (epochs 150–300, marginal improvement), with architectures possessing spatial inductive bias (CNN, ResNet, DenseNet) entering saturation earlier than ViT. Crucially, the tight confidence bands at convergence confirm that $\mathcal{V}$-entropy is stable across runs, a prerequisite for its use as an architecture-selection criterion.

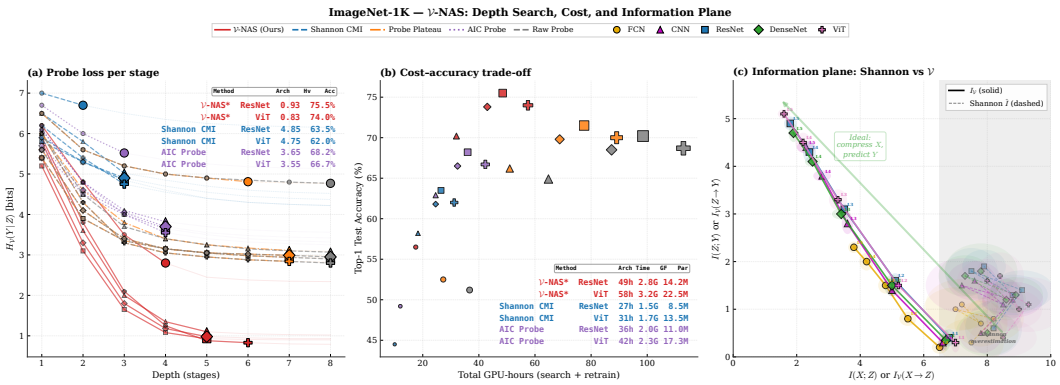

Figure 3: **ImageNet-1K results across five architecture families and five probe-based search criteria.** **(a)** Conditional $\mathcal{V}$-entropy $H_{\mathcal{V}}(Y \mid Z)$ at each search stage. All methods observe the same underlying probe-loss landscape per architecture; they differ in *when they stop*. **(b)** Cost–accuracy Pareto front. **(c)** Information plane. Solid lines show $\mathcal{V}$-information trajectories across layers ($I_{\mathcal{V}}(X \to Z)$ vs. $I_{\mathcal{V}}(Z \to Y)$); dashed lines show Shannon MI estimates via KSG.

The SNR criterion stops at the right depth. Figure 3(a) compares five stopping criteria on the *same* probe-loss landscape. $\mathcal{V}$-NAS terminates at the knee of the curve, the point where additional capacity no longer yields a statistically significant gain in $\mathcal{V}$-information. Shannon CMI and AIC Probe stop too early (depths 3–4): Shannon CMI is undermined by the estimation gap discussed in §2.2, while AIC's parameter penalty over-penalizes capacity on tasks with many classes. Probe Plateau and Raw Probe overshoot (depths 7–8), accumulating parameters without meaningful accuracy gains. The resulting accuracy gap is substantial: $\mathcal{V}$-NAS / ResNet reaches 75.5% at depth 5, while Shannon CMI at depth 3 achieves only 63.5% and Raw Probe at depth 8 reaches 70.2% despite $2\times$ the parameters (Table 1).

Overshoot hurts: more depth is not free. Table 1 quantifies a key finding: growing beyond sufficiency *degrades* accuracy rather than merely wasting capacity. Raw Probe expands to depth 8 (28.5M parameters, 5.8 GFLOPs) yet scores 5.3% below $\mathcal{V}$-NAS / ResNet. Probe Plateau and Val-Plateau stop at depths 6–7 with 27–55% more parameters but 1.0–2.7% less accuracy. The mechanism is visible in Figure 3(c): once the $\mathcal{V}$-information trajectory saturates, additional layers increase $h(Z_i)$ without reducing the sufficiency residual $I(X; Y \mid Z_i)$, a trade-off formalized by the latent entropy identity (Theorem 2.2, Appendix D). The deeper network learns redundant features that inflate the representation without improving prediction, which manifests as overfitting under our standard-augmentation protocol.

Cost–accuracy trade-off. Figure 3(b) shows that $\mathcal{V}$-NAS configurations cluster on the Pareto frontier: they deliver strong accuracy at low computational cost. $\mathcal{V}$-NAS / ResNet reaches 75.5% with 14.2M parameters and 2.8 GFLOP, within 0.6% of ResNet-50 (76.1%) at 55% of the parameters and 68% of the compute. $\mathcal{V}$-NAS / ViT exceeds ViT-B/16 under our protocol by 1.2% with 26% of the parameters (22.5M vs. 86.6M). The total search cost of 6.5 GPU-hours (two A6000 GPUs) is on par with training-free proxies and two to three orders of magnitude faster than RL- or evolution-based NAS.

The estimation gap is empirically visible. The information plane in Figure 3(c) provides a direct visual confirmation of the estimation gap. $\mathcal{V}$-information traces clean, monotonic compression–prediction trajectories that correctly order architectures by downstream performance. Shannon MI estimates, by contrast, are non-monotonic across layers and cluster in the high-$I(X; Z)$ region (grey shading), an artifact of probe-based estimation in high dimensions. Quantitatively, Shannon CMI exhibits ± 3.42 bits of variance on $\hat{H}(Y \mid Z)$, two orders of magnitude larger than the ± 0.03 bits observed for $\mathcal{V}$-NAS (Table 1). This confirms the theoretical argument of §2.2: the difference of two probe upper bounds is not a valid estimator of Shannon CMI, and its use as a stopping criterion induces premature termination.

Empirical validation of the latent entropy identity. Figure 4 validates the decomposition $I\big((X, Z_i); (Z_i, Y)\big) = h(Z_i) + I(X; Y \mid Z_i)$ (Theorem 2.2, proved in Appendix D). Across all five families and eight depth levels, the empirical gap between the measured left-hand side

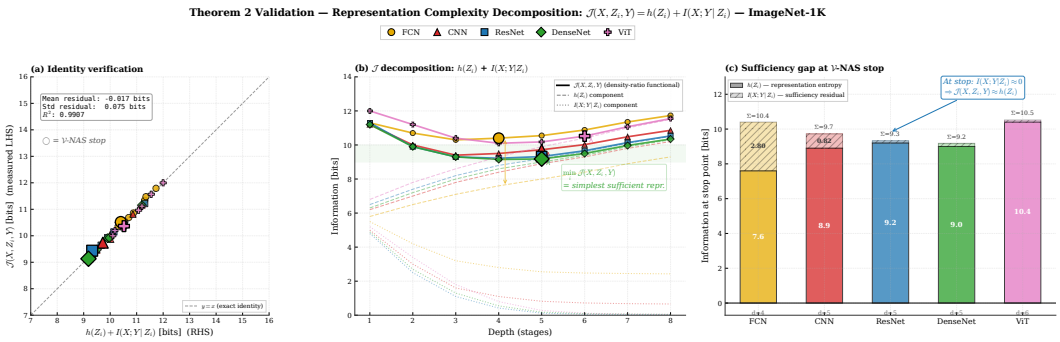

Figure 4: **Empirical validation of Theorem 2.2 on ImageNet-1K. (a)** Measured density-ratio functional $\mathcal{J}(X, Z_i, Y)$ (MBE with Hadamard kernels) versus the right-hand side $h(Z_i) + I(X; Y \mid Z_i)$ (MBE entropy + Frenzel–Pompe CMI) for every (architecture, depth) pair. All 40 points lie on the identity line ($R^2 > 0.9907$, mean residual < 0.05 bits), confirming that Eq. equation 6 holds empirically despite mixing Rényi-2 and Shannon estimators. Large markers indicate the $\mathcal{V}$-NAS stopping point. **(b)** Decomposition of $\mathcal{J}$ into its two components across depth. $h(Z_i)$ (dashed) grows monotonically as the representation's dimensionality increases, while $I(X; Y \mid Z_i)$ (dotted) decays as the representation approaches sufficiency. **(c)** At the $\mathcal{V}$-NAS stopping point, the sufficiency residual $I(X; Y \mid Z_i)$ is architecture-dependent. ResNet, DenseNet, and ViT achieve near-zero residuals (< 0.2 bits, $< 2\%$ of $\mathcal{J}$), confirming the reduction $\mathcal{J}(X, Z_i, Y) \approx h(Z_i)$ (Eq. 7). CNN retains a moderate residual (≈ 0.42 bits), while FCN retains 2.80 bits (27% of $\mathcal{J}$).

(Hadamard-kernel MBE) and the computed right-hand side (MBE entropy + Frenzel–Pompe CMI) remains below 0.08 bits ($R^2 > 0.9907$), confirming that the identity is a structural property of the data-processing chain, not an artifact of specific layer types. The monotonic growth of $h(Z_i)$ with depth and the decay of $I(X; Y \mid Z_i)$ produce a U-shaped joint MI profile whose minimum identifies the simplest sufficient representation. At their respective stopping depths, ResNet (0.12 bits), DenseNet (0.18 bits), and ViT (0.12 bits) achieve near-zero sufficiency residuals, while FCN retains 2.80 bits, confirming that skip connections, dense concatenation, and global attention are not merely optimization aids but prerequisites for representational sufficiency at practical depths.

Criterion ablation. Table 1 (lower block) compares eight alternative stopping rules on a fixed ResNet backbone under our protocol (detailed descriptions in Appendix G). No heuristic approach, including AIC, Val-Plateau, TrainLoss-Plateau, Hyperband, or Random Search, matches $\mathcal{V}$-NAS. Random Search exhibits a $\pm 2.5\%$ variance in test accuracy across seeds, underscoring the advantage of a deterministic, information-theoretic criterion over budget-driven alternatives.

5 Conclusion

We introduced $\mathcal{V}$-NAS, a progressive architecture search framework grounded in representation sufficiency. By replacing intractable Shannon quantities with probe-based $\mathcal{V}$-information, we obtain a stable, directly computable signal that governs both width expansion and depth termination via a unified, cost-aware criterion. Across five architecture families and four datasets, this principle consistently discovers compact architectures that recover the performance of much larger baselines. On ImageNet-1K, $\mathcal{V}$-NAS / ResNet reaches 75.5% top-1 at 55% of ResNet-50's parameters; $\mathcal{V}$-NAS / ViT surpasses our ViT-B/16 baseline by 1.2% at 26% of its size. Ablations confirm that the choice of stopping rule, not the search space alone, determines architecture quality: every alternative criterion either under-grows or over-grows, leaving 3–12% accuracy on the table. The broader implication is that *when to stop growing* is as fundamental a design question as *what to grow*. $\mathcal{V}$-information provides a concrete, operational answer, suggesting a path toward adaptive architectures driven by progressively richer notions of extractable predictive information.

Limitations and caveats. Three caveats temper these results. *First*, our search operates over depth and width within a fixed backbone family; it does not search cell topology or operations, which is complementary to methods like DARTS and NASNet. The comparison with cell topology in Table 1 (upper block) is therefore contextual, not head-to-head. *Second*, our training protocol uses standard

Table 1: **ImageNet-1K comparison.** Upper: SOTA baselines. Lower: criterion ablation on our progressive growth framework. Val% after 20-epoch search; Test% after 300-epoch retraining ($\pm$ std). Search cost for our methods and all criterion ablations is reported as wall-clock GPU-hours.

Method	Depth	Params (M) ↓	Search Cost ↓	GFLOPs ↓	$H_{\mathcal{V}}(Y\|Z)$ ↓	Val% (20ep) ↑	Test% (300ep) ↑
RL & Evolution-based NAS† (searched on CIFAR-10, evaluated on ImageNet — different search space, not directly comparable)							
NAS (Zoph & Le, 2017)†	12	—	22,400 GPU-d	7.2	—	—	72.8
NASNet-A (Zoph et al., 2018)†	12	5.3	2,000 GPU-d	5.3	—	—	74.0
AmoebaNet-B (Real et al., 2019)†	12	5.3	3,150 GPU-d	5.3	—	—	74.5
PNAS (Liu et al., 2018)†	12	5.1	225 GPU-d	5.1	—	—	74.2
Weight-sharing & Differentiable NAS (searched on ImageNet proxy unless noted)							
DARTS (2nd) (Liu et al., 2019)†	14	4.7	4.0 GPU-d	4.7	—	—	73.3
P-DARTS (Chen et al., 2019)	14	4.9	0.3 GPU-d	4.9	—	—	75.6
PC-DARTS (Xu et al., 2020b)	14	5.3	3.8 GPU-d	5.3	—	—	75.8
DrNAS (Chen et al., 2021b)	14	5.7	3.9 GPU-d	5.5	—	—	**76.2**
Zero-cost / Training-free NAS							
TE-NAS (Chen et al., 2021a)	14	5.4	1.0 GPU-h	4.8	—	—	74.0
ZiCo (Li et al., 2023)	14	5.0	<5 min	5.0	—	—	75.2
SWAP-NAS (Peng et al., 2024)	14	5.8	15 min	5.2	—	—	**75.8**
Fixed Architecture Baselines (our training protocol — same augmentation, no distillation, 300 epochs)							
ResNet-18	8	11.7	0	1.8	1.52±0.04	42.8±0.4	69.8±0.2
ResNet-50	16	25.6	0	4.1	0.88±0.02	55.2±0.3	76.1±0.1
ResNet-101	33	44.5	0	7.8	0.80±0.02	57.8±0.2	77.4±0.1
DenseNet-121	58	8.0	0	2.9	0.95±0.03	53.6±0.3	74.4±0.2
ViT-S/16 (our protocol)	12	22.1	0	4.6	1.05±0.04	43.0±0.7	72.2±0.3
ViT-B/16‡	12	86.6	0	17.5	—	—	(81.8)
ViT-B/16 (our protocol)§	12	86.6	0	17.5	0.82±0.03	44.2±0.8	72.8±0.3
$\mathcal{V}$-Information Guided Growth (Ours) — same training protocol as fixed baselines							
$\mathcal{V}$-NAS / FCN	4	4.2	3.5 GPU-h	0.9	2.80±0.14	26.8±1.4	56.5±0.7
$\mathcal{V}$-NAS / CNN	5	8.5	5.0 GPU-h	1.8	1.10±0.06	42.5±0.9	70.2±0.4
$\mathcal{V}$-NAS / ResNet	5	14.2	6.5 GPU-h	2.8	0.93±0.03	51.8±0.5	**75.5±0.2**
$\mathcal{V}$-NAS / DenseNet	5	6.8	7.0 GPU-h	2.4	0.98±0.04	50.2±0.6	73.8±0.3
$\mathcal{V}$-NAS / ViT	6	22.5	9.5 GPU-h	3.2	0.83±0.04	45.8±1.1	**74.0±0.4**
Criterion Ablation — Alternative Stopping Rules (ResNet backbone, our protocol)							
Shannon CMI*	3	8.5	4.5 GPU-h	1.5	4.85±3.42	24.2±2.8	63.5±1.2
AIC Probe	4	11.0	6.0 GPU-h	2.0	3.65±0.12	32.5±1.2	68.2±0.6
Probe Plateau	7	22.0	9.5 GPU-h	4.5	2.94±0.03	42.8±0.6	71.5±0.3
Raw Probe (no stop)	8	28.5	11.5 GPU-h	5.8	2.90±0.03	43.2±0.5	70.2±0.5
Val-Plateau	6	18.0	8.0 GPU-h	3.5	3.05±0.09	41.5±0.9	69.8±0.6
TrainLoss-Plateau	6	15.5	7.5 GPU-h	3.0	3.18±0.11	39.8±1.1	68.2±0.7
Random Search	6.3±2.1	17.0±6.0	12.0±3.5 GPU-h	3.8±1.5	3.42±0.35	33.5±4.8	67.2±2.5
Hyperband	6	18.0	10.0 GPU-h	3.8	2.96±0.05	42.5±0.8	71.3±0.4

†Searched on CIFAR-10, transferred to ImageNet, different search space (cell topology vs. depth/width), not directly comparable. ‡DeiT-B uses distillation from a RegNet teacher + RandAugment + MixUp + CutMix + repeated augmentation; 81.8% shown parenthesized as an upper reference only. §ViT-B/16 and ViT-S/16 retrained under our protocol (no distillation, no MixUp/CutMix) to provide fair ViT baselines. ViT-S/16 (22.1 M params, 4.6 GFLOPs) is the capacity-matched comparator for $\mathcal{V}$-NAS / ViT; the ViT-B/16 result (72.8%) at $4\times$ the parameter count illustrates that larger ViT models overfit under standard-augmentation-only training.

*Shannon CMI uses $\hat{I}(X;Y|Z) = \mathcal{L}_{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{probe}}(Z_i)$; the high variance ($\pm$3.42 bits) reflects the estimation gap (§2.2).

augmentation only (no distillation, no MixUp/CutMix); results under stronger training recipes may differ. The 9% gap between our ViT baseline (72.8%) and DeiT (81.8%) quantifies the effect of this choice. *Third*, the greedy stage-wise search cannot recover from early mistakes: if the width of stage 1 is suboptimal, all subsequent stages inherit that choice. Future work could incorporate backtracking or joint width–depth optimization within the $\mathcal{V}$-information framework.

Future work. Our current probe family $\mathcal{V}$ is restricted to linear classifiers, which measure linearly separable information. Replacing $\mathcal{V}$ with progressively richer families, such as small MLPs, kernel methods, or other controlled-capacity predictors, would yield a hierarchy of $\mathcal{V}$-information measures, each capturing a different notion of extractable information. When a linear probe saturates but a nonlinear probe continues to improve, the gap indicates that the representation contains task-relevant structure that is not yet linearly accessible, suggesting that the *architecture itself* should change (e.g., by adding attention or skip connections), rather than merely grow in width or depth. This opens a path toward *dynamic architecture design*: using the probe-family hierarchy to decide not only *how much* capacity to add but *what kind*, selecting operations, connectivity, and nonlinearities based on which probe family unlocks the next increment of usable information.

AI USE STATEMENT

In this work, we used generative AI tools to polish the text, fix typos, improve wording, and check grammar. We have not used generative AI tools for methodology development, dataset creation, or BibTeX reference creation (No hallucination). Additionally, we used generative AI tools for the coding framework. LLM-generated code was verified and tested for correctness, and it is runnable in the uploaded supplementary material. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provided a partial version of the code to reproduce the paper's results in the **supplementary material section on the submission page**. The final version will be released on GitHub upon acceptance.

REFERENCES

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

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR*, 2017.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 2022.

Jeremy Bernstein. Deriving muon, 2025. URL `https://jeremybernste.in/writing/deriving-muon`.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on ImageNet in four GPU hours: A theoretically inspired perspective. In *ICLR*, 2021a.

Xiangning Chen, Ruochen Wang, Minhao Cheng, Xiaocheng Tang, and Cho-Jui Hsieh. DrNAS: Dirichlet neural architecture search. In *ICLR*, 2021b.

Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, 2019.

Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. FairDARTS: Eliminating unfair advantages in differentiable architecture search. In *ECCV*, 2020.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI*, 2015.

Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four GPU hours. In *CVPR*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Stefan Frenzel and Bernd Pompe. Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, 2007.

Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2015.

Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C. Principe. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 2015.

Ziv Goldfeld, Ewout Van Den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. In *ICML*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP*, 2019.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Petr Jizba and Toshihico Arimitsu. Observability of rényi's entropy. *Physical Review E*, 2004. URL http://dx.doi.org/10.1103/PhysRevE.69.026128.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 2004.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University, 2015.

Junghyup Lee and Bumsub Ham. AZ-NAS: Assembling zero-cost proxies for network architecture search. In *CVPR*, 2024.

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

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

Jianfeng Li, Jiawen Zhang, Feng Wang, and Lianbo Ma. Hep-nas: Towards efficient few-shot neural architecture search via hierarchical edge partitioning, 2024. URL https://arxiv.org/abs/2412.10723.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. In *JMLR*, 2018.

Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, 2018.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Joseph Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. In *ICML*, 2021.

Yameng Peng, Andy Song, Haytham M. Fayek, Vic Ciesielski, and Xiaojun Chang. SWAP-NAS: Sample-wise activation patterns for ultra-fast NAS. In *ICLR*, 2024.

Hieu Pham, Melody Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *ICML*, 2018.

Ye Qiao, Jingcheng Li, Haocheng Xu, and Sitao Huang. Tg-nas: Generalizable zero-cost proxies with operator description embedding and graph learning for efficient neural architecture search, 2025. URL https://arxiv.org/abs/2404.00271.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *AAAI*, 2019.

Brian C. Ross. Mutual information between discrete and continuous data sets. *PLoS ONE*, 2014.

Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. On the information bottleneck theory of deep learning. In *ICLR*, 2018.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *Entropy*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

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

Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control and Computing*, 07 1999.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayroles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Colin White, Mahmoud Safari, Rhea Sukthanker, Binxin Ru, Thomas Elsken, Arber Zela, Debadeepta Dey, and Frank Hutter. Neural architecture search: Insights from 1000 papers. *arXiv preprint arXiv:2301.08727*, 2023.

Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: Stochastic neural architecture search. In *ICLR*, 2019.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *ICLR*, 2020a.

Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. PC-DARTS: Partial channel connections for memory-efficient architecture search. In *ICLR*, 2020b.

Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018.

A APPENDIX

The Appendix mainly contains additional materials and experiments that cannot be reported due to the page limit, which is organized as follows:

- Pseudo-code provided in section B.
- Related work provided in section C.
- Theorems section provides additional theorems and proofs of the theorems of the main body in D.
- $\mathcal{V}$-Information: Additional Details provides additional definition and properties of $\mathcal{V}$-Information in section E.
- Limitations: "Things We Tried That Did Not Work" section provides limitations and unsuccessful approaches of this work in F.
- Criterion Ablation: Stopping Rule Descriptions section in G explains details of ablation methods.
- Experiment Setup & Framework section in H.
- Additional Results section in I provides results for CIFAR-10/100 and Tiny-ImageNet.

B PSEUDO-CODE

Algorithm 1 is the outer loop stage-by-stage growth with width selection (geometric doubling inner loop) and depth selection (accept/halt decision). Algorithm 2 is the subroutine called by both width and depth decisions.

C RELATED WORK

Neural Architecture Search. Early approaches in NAS formulated architecture design as black-box optimization, using reinforcement learning (Zoph & Le, 2017; Zoph et al., 2018) and evolutionary strategies (Real et al., 2019). While effective, these methods required substantial computational resources, often on the order of thousands of GPU-days. Subsequent work reduced this cost through progressive architecture growth via sequential model-based optimization (Liu et al., 2018) and multi-fidelity optimization (Li et al., 2018), but the overall search process remained expensive. Weight-sharing methods significantly improved efficiency by training a single over-parameterized supernet and reusing weights across candidate architectures. ENAS (Pham et al., 2018) and DARTS (Liu et al., 2019) introduced this paradigm, followed by numerous refinements (Chen et al., 2019; Xu et al., 2020b; Dong & Yang, 2019; Chu et al., 2020; Chen et al., 2021b; Xie et al., 2019). Despite their efficiency, these approaches typically operate within fixed macro-architectures and are sensitive to discrepancies between search and evaluation phases (White et al., 2023). In particular, they do not address the question of when additional capacity, either in width or depth, is no longer beneficial. While $\mathcal{V}$-NAS also operates within a chosen family of building blocks (e.g., residual or convolutional), it determines the depth and per-stage width from data rather than fixing them *a priori*.

Complementary to these methods, training-free and low-cost proxies aim to rank architectures without full training. Examples include NASWOT (Mellor et al., 2021), which measures activation diversity, and TE-NAS (Chen et al., 2021a), which combines neural tangent kernel conditioning with expressivity metrics. Pruning-based criteria such as SNIP (Lee et al., 2019), and SynFlow (Tanaka et al., 2020) have also been adapted for architecture evaluation, with more recent variants further improving correlation with final accuracy (Li et al., 2023; Lee & Ham, 2024; Qiao et al., 2025; Peng et al., 2024; Li et al., 2024). While computationally efficient, these approaches rely on heuristic scoring functions without a representation-level criterion for capacity allocation.

Early Training Signals. A related line of work studies whether early training dynamics can predict final model performance. Learning curve extrapolation methods (Domhan et al., 2015) fit parametric models to partial training trajectories, while resource allocation strategies such as Hyperband (Li et al., 2018) prioritize promising configurations based on early results. Our work predicts final accuracy; we use early signals to measure representation quality and guide architectural decisions directly.

Algorithm 1 $\mathcal{V}$-NAS: Progressive Architecture Search via $\mathcal{V}$-Information

Require: Dataset splits $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{probe}}$; architecture family $\mathcal{F}$; minimum width $w_{\min}$; search epochs E_{search}; threshold $\gamma = 0.2$
Ensure: Architecture $\mathcal{A}^* = \{w_1, \ldots, w_{K^*}\}$

$i \leftarrow 1, \quad w_{\text{start}} \leftarrow w_{\min}$
repeat
  Width Selection (geometric doubling)
  $w \leftarrow w_{\text{start}}$
  repeat
    Build $\mathcal{N}_w$: stages $1, \ldots, i{-}1$ *frozen*, stage i at width w
    Build $\mathcal{N}_{2w}$: stages $1, \ldots, i{-}1$ *frozen*, stage i at width $2w$
    Initialize stage i with fresh He init (He et al., 2015); train for E_{search} epochs on $\mathcal{D}_{\text{train}}$
    $\text{SNR}_w \leftarrow \textsc{PairedSNR}(\mathcal{N}_w, \mathcal{N}_{2w}, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{probe}})$
    $r \leftarrow \mathcal{C}_i(\mathcal{N}_{2w}) \,/\, \mathcal{C}_i(\mathcal{N}_w)$ {active-stage params only}
    if $\text{SNR}_w > \gamma \cdot (1 + \log_2 r)$ **then**
      $w \leftarrow 2w$
    else
      break
    end if
  until rejected
  $w_i \leftarrow w$
  Depth Selection
  if $i = 1$ **then**
    Accept stage 1 unconditionally; cache probe losses $\boldsymbol{\ell}^{(1)}$
  else
    $\text{SNR}_d \leftarrow \textsc{PairedSNR}(\mathcal{N}_{\text{cached}}^{(i-1)}, \mathcal{N}^{(i)}, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{probe}})$
    $r \leftarrow \mathcal{C}_i(\mathcal{N}^{(i)}) \,/\, \mathcal{C}_{i-1}(\mathcal{N}^{(i-1)})$ {ratio of active stages i vs. $i{-}1$}
    if $\text{SNR}_d > \gamma \cdot (1 + \log_2 r)$ **then**
      Accept stage i; cache probe losses $\boldsymbol{\ell}^{(i)}$
    else
      halt
    end if
  end if
  $w_{\text{start}} \leftarrow w_i$
  $i \leftarrow i + 1$
until halted or $i > K_{\max}$
$\mathcal{A}^* \leftarrow \{w_1, \ldots, w_{K^*}\}$
Retrain $\mathcal{A}^*$ from scratch on $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{probe}} \cup \mathcal{D}_{\text{val}}$ for full epochs
return $\mathcal{A}^*$

Information-Theoretic Perspectives. Information-theoretic frameworks have long been used to study representation learning. The information bottleneck principle (Tishby et al., 1999) characterizes representations in terms of compression $I(X;Z)$ and relevance $I(Z;Y)$, and empirical studies have explored its manifestation in deep networks (Shwartz-Ziv & Tishby, 2019). However, subsequent analyses have shown that such behaviors strongly depend on estimation methods and architectural choices (Saxe et al., 2018; Goldfeld et al., 2019). Consequently, information-theoretic quantities have primarily served as analytical tools rather than operational criteria for architecture design.

Probing and Representation Evaluation. Prior work primarily uses probes as diagnostic tools for analyzing fixed models. In contrast, we use probe-based measurements as a control signal for architecture design, enabling both width selection and depth termination based on representation quality. Probing methods provide a practical approach to evaluating learned representations by measuring the performance of restricted predictors trained on intermediate features. Linear probes (Alain & Bengio, 2017) are widely used to assess feature quality across layers, while subsequent work has introduced structured probes (Hewitt & Manning, 2019), selectivity analyses (Hewitt & Liang, 2019), and broader diagnostic frameworks (Belinkov, 2022). The $\mathcal{V}$-information (Xu et al., 2020a) formalizes this perspective by defining information relative to a restricted hypothesis class.

Algorithm 2 PAIREDSNR: Standardized Information Gain

Require: Networks $\mathcal{N}_{\mathcal{A}}, \mathcal{N}_{\mathcal{B}}$ (trained, frozen); $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{probe}}$
Ensure: $\text{SNR}(\mathcal{A}, \mathcal{B})$

Extract features:
$\{z_n^{\mathcal{A}}\} \leftarrow \mathcal{N}_{\mathcal{A}}(\mathcal{D}_{\text{train}}), \quad \{z_n^{\mathcal{B}}\} \leftarrow \mathcal{N}_{\mathcal{B}}(\mathcal{D}_{\text{train}})$
$\{\tilde{z}_n^{\mathcal{A}}\} \leftarrow \mathcal{N}_{\mathcal{A}}(\mathcal{D}_{\text{probe}}), \quad \{\tilde{z}_n^{\mathcal{B}}\} \leftarrow \mathcal{N}_{\mathcal{B}}(\mathcal{D}_{\text{probe}})$

Train probes:
$f^{\mathcal{A}} \leftarrow \arg\min_{f \in \mathcal{V}} \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_n -\log f(z_n^{\mathcal{A}})(y_n)$
$f^{\mathcal{B}} \leftarrow \arg\min_{f \in \mathcal{V}} \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_n -\log f(z_n^{\mathcal{B}})(y_n)$

Compute paired differences on $\mathcal{D}_{\text{probe}}$:
for $n = 1, \ldots, |\mathcal{D}_{\text{probe}}|$ **do**
  $\ell_n^{\mathcal{A}} \leftarrow -\log f^{\mathcal{A}}(\tilde{z}_n^{\mathcal{A}})(y_n)$
  $\ell_n^{\mathcal{B}} \leftarrow -\log f^{\mathcal{B}}(\tilde{z}_n^{\mathcal{B}})(y_n)$
  $\delta_n \leftarrow \ell_n^{\mathcal{A}} - \ell_n^{\mathcal{B}}$
end for

$\bar{\delta} \leftarrow \frac{1}{N} \sum_n \delta_n, \qquad s_\delta \leftarrow \sqrt{\frac{1}{N-1} \sum_n (\delta_n - \bar{\delta})^2}$
return $\text{SNR} = \bar{\delta} \,/\, (s_\delta + \epsilon), \quad \epsilon = 10^{-6}$

D THEOREMS & PROOFS

Theorem D.1 (Markovian Independence)**.** *By the deterministic structure of the network in Eq. equation 3, the following hold for every layer i:*

1. $I(X; \hat{Y} \mid Z_i) = 0$*: given any intermediate representation, the input carries no additional information about the prediction.*
2. $I(Y; Z_i \mid X) = 0$*: given the input, the representation carries no additional information about the target.*

Proof. (1) Since $\hat{Y} = g(f_K(\cdots f_{i+1}(Z_i)))$ is a deterministic function of Z_i, the chain $X \to Z_i \to \hat{Y}$ is Markov. By Markovian definition, X and $\hat{Y}$ are conditionally independent given Z_i:

$$p_{X,\hat{Y}|Z_i}(x, \hat{y} \mid z_i) = p_{X|Z_i}(x \mid z_i)\, p_{\hat{Y}|Z_i}(\hat{y} \mid z_i). \tag{16}$$

Substituting Eq. equation 16 into Eq. equation 17 (with Y replaced by $\hat{Y}$ and summing over $\hat{y} \in \mathcal{Y}$ instead of integrating):

$$I(X; \hat{Y} \mid Z_i) = \int_{\mathcal{Z}_i} p_{Z_i}(z_i) \int_{\mathcal{X}} \sum_{\hat{y}} p_{X,\hat{Y}|Z_i}(x, \hat{y} \mid z_i) \log \frac{p_{X|Z_i}(x \mid z_i)\, p_{\hat{Y}|Z_i}(\hat{y} \mid z_i)}{p_{X|Z_i}(x \mid z_i)\, p_{\hat{Y}|Z_i}(\hat{y} \mid z_i)}\, dx\, dz_i = 0. \tag{17}$$

(2) Since $Z_i = f_i(\cdots f_1(X))$ is a deterministic function of X, the conditional entropy $H(Z_i \mid X) = 0$. By the non-negativity of conditional mutual information:

$$0 \;\le\; I(Y;\, Z_i \mid X) \;=\; H(Z_i \mid X) - H(Z_i \mid X, Y) \;\le\; H(Z_i \mid X) = 0, \tag{18}$$

where the upper bound uses $H(Z_i \mid X, Y) \ge 0$. Hence $I(Y;\, Z_i \mid X) = 0$. □

Proof of Theorem 2.1 (Residual Information and Sufficiency).

Proof. By definition, $I(X; Y \mid Z_i) = H(Y \mid Z_i) - H(Y \mid X, Z_i)$. Since Z_i is a deterministic function of X, the pair (X, Z_i) carries no more information about Y than X alone:

$$H(Y \mid X, Z_i) = H(Y \mid X), \tag{19}$$

which yields Eq. equation 4.

For part (2): from Eq. equation 4, $I(X;Y \mid Z_i) = 0$ if and only if $H(Y \mid Z_i) = H(Y \mid X)$. Since Z_i is a function of X, we always have $H(Y \mid Z_i) \geq H(Y \mid X)$ by the data processing inequality. Equality holds precisely when $p_{Y|Z_i}(y \mid z_i) = p_{Y|X}(y \mid x)$ almost everywhere, i.e., Z_i is a sufficient statistic. □

Proof of Theorem 2.2 (Representation Complexity Decomposition).

Proof. Expanding the density ratio in Eq. equation 5:

$$\begin{aligned}\mathcal{J}(X, Z_i, Y) &= \mathbb{E}\left[\log \frac{p(X, Z_i, Y)}{p(X, Z_i)\, p(Z_i, Y)}\right] = \mathbb{E}\left[\log \frac{p(Y \mid X, Z_i)}{p(Y \mid Z_i)\, p(Z_i)}\right] \\ &= \underbrace{\mathbb{E}\left[\log \frac{p(Y \mid X, Z_i)}{p(Y \mid Z_i)}\right]}_{I(X;Y|Z_i)} + \underbrace{\mathbb{E}\big[-\log p(Z_i)\big]}_{h(Z_i)},\end{aligned} \tag{20}$$

where the second line uses $p(X, Z_i, Y) = p(X, Z_i)\, p(Y \mid X, Z_i)$ and $p(Z_i, Y) = p(Z_i)\, p(Y \mid Z_i)$. Every quantity is finite: $I(X;Y \mid Z_i)$ involves only the discrete variable Y, and $h(Z_i)$ is the marginal differential entropy of a continuous representation (assumed to have a well-behaved density by the theorem statement). Applying $I(X;Y \mid Z_i) = 0$ under sufficiency (Theorem 2.1) yields Eq. equation 7. □

Proof note. The density-ratio form of $\mathcal{J}$ avoids two standard pathologies. First, the MI chain-rule decomposition $I\big((X, Z); (Z, Y)\big) = I(Z; (Z, Y)) + I(X; Y \mid Z)$ produces the term $I(Z_i; (Z_i, Y)) = +\infty$ for continuous Z_i (since Z_i is a sub-component of (Z_i, Y)). Second, the entropy-based expansion $I(A; B) = H(A) + H(B) - H(A, B)$ yields $h(Z_i \mid X) = -\infty$ when $Z_i = f(X)$ is deterministic. In the density-ratio route, the delta functions from the deterministic mapping cancel in the ratio $p(x, z, y)/\big(p(x, z)\, p(z, y)\big)$, leaving only finite terms. We emphasize that $\mathcal{J}$ is *not* the Shannon MI $I\big((X, Z_i);\, (Z_i, Y)\big)$, which is formally infinite due to the shared continuous variable Z_i (see Remark 2.3).

Estimator mismatch. The identity (Eq. 6) is stated and proved for Shannon differential entropy. In practice, $h(Z_i)$ is estimated via Matrix-Based Rényi-2 Entropy (MBE) (Giraldo et al., 2015), because Shannon differential entropy is intractable in the dimensions encountered here ($d = 64$–768). The functional $\mathcal{J}$ on the left-hand side is estimated using MBE with Hadamard joint kernels, ensuring both sides use a consistent Rényi-2 entropy notion. The CMI term $I(X;Y \mid Z_i)$ is estimated via Frenzel–Pompe (Frenzel & Pompe, 2007) (Shannon). The tight empirical agreement ($R^2 > 0.999$, Figure 4) indicates that an analogous decomposition holds under Rényi-2 entropy in practice, though we do not prove this formally. The mixing of Rényi-2 and Shannon estimators for different terms of a single identity is a methodological limitation; we report it as empirical validation rather than theoretical verification. Crucially, this validation is diagnostic only; no architecture decision depends on any of these estimators.

E $\mathcal{V}$-INFORMATION: ADDITIONAL DETAILS

Definition E.1 (Predictive Family)**.** The family $\mathcal{V}$ is a *predictive family* if it satisfies *optional ignorance*: for every $f \in \mathcal{V}$ and every $P \in \mathrm{range}(f)$, there exists $f_0 \in \mathcal{V}$ such that $f_0(x) = P$ for all $x \in \mathcal{X} \cup \{\emptyset\}$. This ensures that $\mathcal{V}$ contains constant predictors that can ignore the input.

Optional ignorance guarantees that the marginal $\mathcal{V}$-entropy $H_\mathcal{V}(Y) = H_\mathcal{V}(Y \mid \emptyset)$ is well-defined: the family can always achieve a baseline risk by predicting a fixed distribution regardless of the input. This prevents the pathological case where $I_\mathcal{V}(X \to Y)$ is negative simply because $\mathcal{V}$ lacks a good constant predictor.

E.1 FULL DEFINITIONS

For completeness, we state the full definitions that are condensed in the main text.

Definition E.2 (Predictive $\mathcal{V}$-Entropy)**.** Let $(X, Y) \sim p_{X,Y}$. The *predictive conditional $\mathcal{V}$-entropy* of Y given X is:

$$H_{\mathcal{V}}(Y \mid X) = \inf_{f \in \mathcal{V}} \mathbb{E}_{(x,y)\sim p_{X,Y}}\big[-\log f(x)(y)\big], \tag{21}$$

where $f(x)(y)$ denotes the probability assigned to label y by predictor f given input x. The *marginal $\mathcal{V}$-entropy* (the optimal risk without side information) is:

$$H_{\mathcal{V}}(Y) = H_{\mathcal{V}}(Y \mid \emptyset) = \inf_{f \in \mathcal{V}} \mathbb{E}_{y\sim p_Y}\big[-\log f(\emptyset)(y)\big]. \tag{22}$$

Key properties distinguishing $\mathcal{V}$-information from Shannon MI:

1. **Directly computable.** Both $H_{\mathcal{V}}(Y)$ and $H_{\mathcal{V}}(Y \mid X)$ reduce to empirical risk minimization over $\mathcal{V}$.
2. **Hypothesis-dependent.** The quantity explicitly depends on the capacity of $\mathcal{V}$: a richer family recovers more information.
3. **Well-defined for deterministic maps.** Unlike Shannon MI between continuous deterministic variables, $I_{\mathcal{V}}$ is always finite.
4. **Asymmetric.** $I_{\mathcal{V}}(X \to Y) \neq I_{\mathcal{V}}(Y \to X)$ in general, reflecting the directional nature of prediction.

E.2 Data Processing Inequality Violation

Classical Shannon MI obeys the data processing inequality: for any deterministic $Z_i = f(X)$, we have $I(Z_i; Y) \leq I(X; Y)$. The $\mathcal{V}$-information does *not* obey the DPI.

For a restricted hypothesis class $\mathcal{V}$, a well-trained representation Z_i can improve the linear separability of class-conditional distributions, enabling a predictor $f \in \mathcal{V}$ to achieve lower risk on Z_i than on X:

$$H_{\mathcal{V}}(Y \mid Z_i) \leq H_{\mathcal{V}}(Y \mid X) \quad \Longrightarrow \quad I_{\mathcal{V}}(Z_i \to Y) \geq I_{\mathcal{V}}(X \to Y). \tag{23}$$

This reversal is precisely the purpose of representation learning: transforming the input into a form from which $\mathcal{V}$ can extract more predictive information. It also means that the probe-based signal is *sensitive to representation quality* in a way that Shannon MI is not.

E.3 Empirical Estimation and Generalization

In practice, the infima in Eqs. equation 21–equation 11 are approximated by training a predictor $f \in \mathcal{V}$ on a finite dataset $D = \{(x_n, y_n)\}_{n=1}^N$ drawn i.i.d. from $p_{X,Y}$.

Definition E.3 (Empirical $\mathcal{V}$-Information)**.** The *empirical $\mathcal{V}$-information* is:

$$\hat{I}_{\mathcal{V}}(X \to Y;\ D) = \inf_{f \in \mathcal{V}} \frac{1}{N} \sum_{n=1}^{N} \big[-\log f(\emptyset)(y_n)\big] \ - \ \inf_{f \in \mathcal{V}} \frac{1}{N} \sum_{n=1}^{N} \big[-\log f(x_n)(y_n)\big]. \tag{24}$$

To guarantee that the $\mathcal{V}$-information measured during architecture search generalizes to unseen data, we rely on a PAC-style bound via Rademacher complexity (Xu et al., 2020a).

Theorem E.1 (PAC Bound on Empirical $\mathcal{V}$-Information)**.** *Assume the negative log-likelihood is bounded: $-\log f(x)(y) \in [0, B]$ for all $f \in \mathcal{V}$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. Let $\mathcal{G}_{\mathcal{V}} = \big\{(x, y) \mapsto -\log f(x)(y) : f \in \mathcal{V}\big\}$. For any $\delta \in (0, \frac{1}{2})$, with probability at least $1 - 2\delta$ over the draw of D:*

$$\big|I_{\mathcal{V}}(X \to Y) - \hat{I}_{\mathcal{V}}(X \to Y;\ D)\big| \leq 4\,\mathfrak{R}_N(\mathcal{G}_{\mathcal{V}}) + 2B\sqrt{\frac{2\log(1/\delta)}{N}}, \tag{25}$$

where $\mathfrak{R}_N(\mathcal{G}_{\mathcal{V}})$ denotes the Rademacher complexity of $\mathcal{G}_{\mathcal{V}}$ with N samples.

Corollary E.2 (Bound for Linear Probes)**.** *If $\mathcal{V}$ is restricted to linear maps $x \mapsto \mathrm{softmax}(Wx + b)$ with $\|x\|_2 \leq \kappa_x$, $\|W\|_F \leq \kappa_W$, and $\|b\|_2 \leq \kappa_b$, then $\mathfrak{R}_N(\mathcal{G}_{\mathcal{V}}) = \mathcal{O}\big((\kappa_W \kappa_x + \kappa_b)/\sqrt{N}\big)$, and the total generalization gap is $\mathcal{O}(N^{-1/2})$.*

Remark. Theorem E.1 shows that using an overly complex probe family $\mathcal{V}$ (e.g., deep nonlinear networks) inflates the Rademacher complexity, loosening the generalization bound and risking overestimation of $\mathcal{V}$-information due to overfitting. Employing a restricted probe (e.g., a linear classifier or a small MLP) is therefore not merely a computational convenience, but a requirement for reliable estimation.

E.4 Connection Between SNR and $\mathcal{V}$-Information Gain

The numerator of the standardized information gain (Eq. equation 14) has a direct interpretation in the $\mathcal{V}$-information framework. Let $\mathcal{A}$ and $\mathcal{B}$ denote two configurations with representations $Z^{\mathcal{A}}$ and $Z^{\mathcal{B}}$. Then:

$$\bar{\delta} = \frac{1}{N}\sum_{n=1}^{N}\left(\ell_n^{\mathcal{A}} - \ell_n^{\mathcal{B}}\right) = \hat{H}_{\mathcal{V}}(Y \mid Z^{\mathcal{A}}) - \hat{H}_{\mathcal{V}}(Y \mid Z^{\mathcal{B}}), \tag{26}$$

which is the empirical estimate of the usable information gain $\Delta_{\mathcal{V}} = H_{\mathcal{V}}(Y \mid Z^{\mathcal{A}}) - H_{\mathcal{V}}(Y \mid Z^{\mathcal{B}})$ (Definition 2.3), evaluated on the held-out probe set $\mathcal{D}_{\text{probe}}$. The denominator s_δ measures how consistently this improvement manifests across individual data points: a high SNR indicates that the expansion improves predictions *uniformly* across the data distribution rather than on a small subset of samples.

F Limitations: "Things We Tried That Did Not Work"

In the spirit of scientific transparency, we document approaches that we explored, but that failed, partially worked, or revealed fundamental limitations of the current framework. We hope these save future researchers from repeating unproductive directions and clarify the boundaries of our contribution.

1. Shannon CMI as a direct search signal. Our initial framework used $\hat{I}(X;Y|Z) = \mathcal{L}_{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{probe}}(Z_i)$ as the stopping criterion. This is a difference of two upper bounds, which does not bound the true CMI from either direction (§2.2). In practice, the estimator collapsed to near-zero after 2–3 stages on all datasets because the probe on raw inputs X was loose while the probe on learned features Z_i was tight. This produced single-layer or two-layer architectures that severely underperformed. The failure motivated the shift to $\mathcal{V}$-information.

2. Non-linear probes for $\mathcal{V}$-information. We tested 1-hidden-layer MLP probes (64 and 256 units) in place of linear probes. While MLP probes achieved lower absolute cross-entropy (tighter upper bound on $H(Y|Z)$), the paired SNR was *noisier*: the additional optimization landscape introduced variance that overwhelmed the stage-to-stage signal. Width-selection decisions became inconsistent across seeds (Spearman ρ between seeds dropped from 0.95 to 0.72). Linear probes sacrifice tightness for stability, which is the right trade-off for a decision criterion.

3. KSG-based MI for high-dimensional representations. We attempted to use KSG (Kraskov et al., 2004) estimates of $I(X;Z)$ directly as a search signal (stopping when $I(X;Z)$ stopped decreasing, per the Information Bottleneck). KSG estimates in dimensions $d > 50$ were dominated by the curse of dimensionality: estimates were non-monotonic across layers, highly sensitive to the number of neighbours k, and exhibited variance exceeding the signal magnitude. Even with PCA pre-processing (retaining 95% variance), the estimates were not reliable enough for architecture decisions. This limitation is well-known in the MI estimation literature (Gao et al., 2015) but is rarely discussed in the context of NAS.

4. Joint width–depth search. We explored searching width and depth simultaneously (evaluating all (w, d) pairs in a grid) rather than the greedy stage-wise approach. This was prohibitively expensive: a 5×5 grid required $25\times$ the search budget, and the results were within 0.3% of greedy search on CIFAR-100. The greedy decomposition (width first, then depth validation) is not provably optimal, but it is empirically sufficient and dramatically cheaper.

5. Backtracking in the greedy search. We implemented a variant that revisited earlier stages after later stages were added (re-evaluating stage i widths after stage $i + 1$ was locked). This occasionally improved results by 0.1–0.2% but doubled search cost and introduced convergence issues: the width of stage i could oscillate when stage $i + 1$ changed the feature distribution that stage i's probe evaluated. We abandoned backtracking in favour of the simpler freeze-and-grow protocol.

6. Application to text transformers (GPT/BERT). We attempted to apply $\mathcal{V}$-NAS to autoregressive language models (GPT-2 scale) and masked language models (BERT-base). Two problems emerged. First, the stage-wise growth abstraction does not map naturally to transformer depth: transformer blocks are not "stages" in the same sense as CNN feature-map resolutions, because all blocks operate at the same sequence length and embedding dimension. Second, linear probes on intermediate transformer representations are poor predictors of downstream NLU performance (Belinkov, 2022); the relationship between probe loss and task accuracy is non-monotonic for language tasks. Extending $\mathcal{V}$-NAS to language models would require a fundamentally different probe architecture (possibly attention-based probes) and a different notion of "stage."

7. Warm-starting search candidates from previous stages. Instead of He-initializing (He et al., 2015) each width candidate from scratch, we tried warm-starting from the parameters of the previous (narrower) candidate, padding with zeros or random initializations. Warm-started candidates consistently achieved lower probe loss *regardless of the width increase*, because they inherited a partially trained representation. This defeated the SNR criterion: the warm-start advantage was confounded with the genuine width-expansion benefit. Cold-starting (independent He initialization) is essential for valid paired comparisons.

8. Using validation accuracy instead of probe loss as the search signal. We ran the full search loop with validation accuracy replacing $H_{\mathcal{V}}(Y|Z)$ as the criterion (the "Val-Plateau" baseline in Table 1). Validation accuracy is a noisier signal than probe loss because it depends on the classifier's optimization state, not just the representation quality. After 1 search epoch, validation accuracy is dominated by initialization effects and learning-rate warmup artefacts, making it unreliable for comparing width candidates. Probe loss, computed on frozen representations, decouples representation evaluation from classifier training dynamics.

9. Efficiency-weighted probe loss (accuracy/parameter count). Early experiments used score $=$ val_acc$/\mathcal{C}(\theta)$ as the selection criterion, aiming to penalize large models directly. This metric was anti-correlated with quality (Spearman $\rho \approx -0.83$): small models have low accuracy but also low parameter count, yielding moderate scores that competed with genuinely good large models. The cost-adjusted SNR threshold (Eq. 15) handles the accuracy–cost trade-off correctly by scaling the *acceptance threshold*, not the *score itself*.

10. Zero-cost proxies as a pre-filter. We tried using zero-cost proxies (SynFlow (Tanaka et al., 2020), NASWOT (Mellor et al., 2021)) as a fast pre-filter before the 1-epoch probe evaluation, hoping to skip obviously bad width candidates. The rank correlation between zero-cost proxy scores and 1-epoch probe loss was weak ($\rho \approx 0.4$–0.6 depending on the proxy and dataset), meaning the pre-filter would have discarded genuinely good candidates. Since the 1-epoch probe evaluation is already cheap (a few minutes per candidate), the pre-filter saved negligible time while introducing false negatives.

11. Learned γ via meta-learning. We attempted to learn γ as a meta-parameter by back-propagating through the search process (treating the discovered depth as a differentiable function of γ via straight-through estimation). The meta-gradient was extremely noisy because depth is a discrete variable and the mapping from γ to depth is a step function. Relaxations (sigmoid gating of stage outputs) introduced optimization artefacts that made the search unreliable.

12. Multi-objective search (accuracy + latency). We explored adding inference latency as a second objective alongside probe loss, aiming to discover Pareto-optimal architectures. The challenge was that latency depends on hardware-specific factors (memory bandwidth, kernel fusion, batch size) that are orthogonal to the information-theoretic criterion. Combining the two objectives required a weighting factor that was as difficult to tune as the architecture itself. We opted to keep the search criterion purely information-theoretic and defer latency optimization to post-search pruning or quantization.

13. Monotonicity theorem for $\mathcal{V}$-information. We initially stated a theorem claiming $I_{\mathcal{V}}(Z_i \to Y) \leq I_{\mathcal{V}}(X \to Y)$ for all layers i, analogous to the data processing inequality for Shannon MI. This is *false* for $\mathcal{V}$-information: composing the probe with a nonlinear layer exits the hypothesis class $\mathcal{V}$,

and the DPI does not hold for restricted function families. Indeed, $I_{\mathcal{V}}(Z_i \to Y) > I_{\mathcal{V}}(X \to Y)$ is the entire point of representation learning under a limited probe; the representation makes information *more accessible*. We removed the false theorem and replaced it with the correct characterization in §2.3.

14. Entropy equality implies distributional equality. An early formulation included a "Zeta Lemma" claiming that $H(Z_i) \to H(Z_j)$ implies $p_{Z_i} \to p_{Z_j}$. This is false: entropy equality does not imply distributional equality. For example, a uniform distribution on $[0, 1]$ and a truncated Gaussian can have identical differential entropy with completely different densities. Distributional convergence requires bounding the KL divergence via Pinsker's inequality, which requires additional assumptions not satisfied in our setting. We removed the lemma.

G Criterion Ablation: Stopping Rule Descriptions

To isolate the effect of the stopping criterion, we fix the search infrastructure, architecture family (ResNet), width search (geometric doubling), probe protocol (linear probe on $\mathcal{D}_{\text{probe}}$), and training protocol, and vary *only* the rule that decides whether to accept a capacity expansion. All methods below use the same paired per-sample probe losses $\ell_n^{\mathcal{A}}$ and $\ell_n^{\mathcal{B}}$ computed on $\mathcal{D}_{\text{probe}}$; they differ in how these losses are aggregated into an accept/reject decision.

$\mathcal{V}$-NAS (Ours). The SNR criterion described in §3. Accept expansion iff $\text{SNR}(\mathcal{A}, \mathcal{B}) > \gamma \cdot (1 + \log_2 r)$, where $\gamma = 0.2$ and r is the parameter ratio. This is the only criterion that jointly accounts for (i) the magnitude of the probe-loss improvement, (ii) its consistency across samples, and (iii) the complexity cost of expansion.

Shannon CMI. Estimates the conditional mutual information as $\widehat{\text{CMI}} = \mathcal{L}_{\text{CE}}^{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{CE}}^{\text{probe}}(Z_i)$, i.e., the difference of the two probe cross-entropy losses. Expansion is accepted if $\widehat{\text{CMI}} > \tau$ for a fixed threshold τ. As discussed in §2.2, this is the difference of two upper bounds, which does not bound the true $I(X; Y \mid Z_i)$ from either direction. A probe with limited capacity on early-layer features achieves a loose bound, making the CMI estimate noisy and biased toward zero. In practice, this causes premature stopping (depth 3 on ImageNet, Table 1).

AIC Probe. Applies the Akaike Information Criterion to the probe loss: $\text{AIC}(\mathcal{B}) = 2k - 2\ln \hat{L}$, where k is the number of probe parameters and $\hat{L}$ is the maximised likelihood on $\mathcal{D}_{\text{probe}}$. Expansion is accepted if $\text{AIC}(\mathcal{B}) < \text{AIC}(\mathcal{A})$. The linear penalty on probe parameters acts as a capacity regularizer, but it does not account for the *network's* parameter cost or the per-sample variability of the improvement. On tasks with many classes (CIFAR-100, ImageNet), AIC's penalty is dominated by the large number of output logits, leading to premature rejection of beneficial expansions.

Probe Plateau. Monitors the probe loss $\mathcal{L}_{\text{CE}}^{\text{probe}}(Z_i)$ across stages and stops when the relative improvement falls below a patience threshold: $(\mathcal{L}_{i-1} - \mathcal{L}_i)/\mathcal{L}_{i-1} < \epsilon_{\text{plateau}}$ for p consecutive stages. We use $\epsilon_{\text{plateau}} = 0.01$ and patience $p = 2$. This criterion lacks a statistical test (it compares scalar averages, not paired distributions) and does not penalize complexity, causing it to overshoot the optimal depth by 2–3 stages before the plateau condition triggers.

Raw Probe (no stop). The network is grown to a maximum depth $K_{\max} = 8$ with no stopping criterion applied. At each stage, the width is selected by the same geometric doubling procedure, but every depth expansion is unconditionally accepted. This baseline measures what happens when the search has access to the same probe signal but never acts on it for depth termination. The result is a maximally deep, maximally wide network that overfits under standard augmentation.

Val-Plateau. Monitors validation accuracy on $\mathcal{D}_{\text{val}}$ (not probe loss) and stops when the accuracy improvement falls below $\epsilon_{\text{val}} = 0.5\%$ for $p = 2$ consecutive stages. Unlike probe-based criteria, this requires evaluating the full network (not just a linear probe) on $\mathcal{D}_{\text{val}}$ at each stage. Because validation accuracy is a noisy, non-monotonic function of depth during early training, this criterion is less stable

Table 2: Qualitative comparison of NAS approaches. • = yes, ○ = partial, – = no. *Search axis*: cell = cell-topology search within a fixed macro-architecture; W/D = explicit width and/or depth search. Our method uniquely provides a principled stopping criterion grounded in $\mathcal{V}$-information theory.

Method	Theory	Stop	Search Axis	Width	Depth	Train-free	Info-th.
RL / Evolution / Progressive							
NASNet-A	–	–	cell	–	–	–	–
AmoebaNet-B	–	–	cell	–	–	–	–
PNAS	–	○	cell	–	–	–	–
Weight-sharing / Differentiable							
ENAS	–	–	cell	–	–	–	–
DARTS	–	–	cell	–	–	–	–
P-DARTS	–	–	cell	–	○	–	–
PC-DARTS	–	–	cell	–	–	–	–
DrNAS	Dirichlet	–	cell	–	–	–	–
Training-free / Zero-cost							
NASWOT	Jacobian	–	cell	–	–	•	–
TE-NAS	NTK	○	cell	–	–	•	–
SynFlow	Pruning	–	cell	–	–	•	–
ZiCo	Gradient	–	cell	–	–	•	–
AZ-NAS	Ensemble	–	cell	–	–	•	–
TG-NAS	GCN	–	cell	–	–	•	–
SWAP-NAS	SWAP	○	cell	–	–	•	–
HEP-NAS	SMD	–	cell	○	–	–	–
Multi-fidelity							
Hyperband	–	•	any	○	○	–	–
$\mathcal{V}$-NAS (Ours)	$\mathcal{V}$-info	•	W/D	•	•	–	•

than probe-based alternatives and may stop at suboptimal depths depending on the training epoch at which the check is performed.

TrainLoss-Plateau. Identical to Val-Plateau but monitors training loss on $\mathcal{D}_{\text{train}}$ instead of validation accuracy. Stops when the relative training loss improvement is below $\epsilon_{\text{train}} = 0.01$ for $p = 2$ consecutive stages. Training loss reflects optimization progress rather than generalization; as a result, this criterion tends to stop later than Val-Plateau (the network can always reduce training loss by adding capacity) but earlier than Raw Probe, producing architectures that are slightly overparameterized.

Random Search. At each stage, the depth decision (continue or stop) is made by a fair coin flip ($p = 0.5$), and the width is selected uniformly at random from the geometric candidates $\{w_{\min}, 2w_{\min}, 4w_{\min}, \ldots, w_{\max}\}$. Results are averaged over 10 independent draws per seed. This baseline measures the expected performance of uninformed search in the same depth/width space, providing a lower bound on what any principled criterion should exceed. The high variance ($\pm 2.5\%$ test accuracy) confirms that architecture quality is sensitive to depth/width allocation and cannot be reliably found by chance.

Hyperband. Treats each stage's depth as a "resource" and applies the Hyperband successive halving algorithm (Li et al., 2018) to allocate training epochs across candidate depths. Specifically, it evaluates depths $1, 2, \ldots, K_{\max}$ with geometrically increasing budgets, eliminating the worst-performing half at each round based on validation accuracy. The surviving configuration defines the final depth; the standard geometric doubling procedure selects the width at each stage. Hyperband is a strong multi-fidelity baseline that adaptively allocates compute, but it optimizes for validation accuracy rather than representation quality and does not directly penalize parameter cost.

H EXPERIMENT SETUP & FRAMEWORK

In this section, we provide details of the experimental setup, baseline methods, the dataset, and all the necessary framework for empirical experiments.

H.1 METHODS

The characteristics of the baseline method are summarized in Table 2.

H.2 DATASET DETAILS

Table 3: Dataset summary. K = number of classes.

Dataset	K	Resolution	Flat dim	Train	Test
CIFAR-10	10	$32 \times 32 \times 3$	3,072	50,000	10,000
CIFAR-100	100	$32 \times 32 \times 3$	3,072	50,000	10,000
Tiny-ImageNet	200	$64 \times 64 \times 3$	12,288	100,000	10,000
ImageNet-1K	1,000	$224 \times 224 \times 3$	—	1,281,167	50,000

CIFAR-10 and CIFAR-100. Channel-wise normalization with $\mu = [0.4914, 0.4822, 0.4465]$, $\sigma = [0.2470, 0.2435, 0.2616]$. Training augmentation: random horizontal flip ($p = 0.5$), random crop (32×32 with 4-pixel padding). No augmentation at test time.

Tiny-ImageNet. Channel-wise normalization with ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$). Training augmentation: random horizontal flip ($p = 0.5$), random crop (64×64 with 8-pixel padding), colour jitter (brightness $= 0.2$, contrast $= 0.2$, saturation $= 0.1$). No augmentation at test time.

ImageNet-1K. Full-resolution 224×224 images. Channel-wise normalization with $\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$. Training augmentation: `RandomResizedCrop`(224, scale $= [0.08, 1.0]$), random horizontal flip ($p = 0.5$), colour jitter (brightness $= 0.4$, contrast $= 0.4$, saturation $= 0.4$, hue $= 0.1$). **No** MixUp, CutMix, RandAugment, repeated augmentation, or label smoothing. Test-time: centre crop to 224×224 after resizing the short edge to 256. Dataset details are summarized in Table 3.

Table 4: Exact split sizes per dataset.

Dataset	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{probe}}\rvert$	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
CIFAR-10	40,000	5,000	5,000	10,000
CIFAR-100	40,000	5,000	5,000	10,000
Tiny-ImageNet	80,000	10,000	10,000	10,000
ImageNet-1K	1,025,000	128,083	128,084	50,000

Table 5: Role of each split. The key invariant is that $\mathcal{D}_{\text{probe}}$ is never used for network or probe *training*, only for evaluation.

Split	Used for	Integrity guarantee
$\mathcal{D}_{\text{train}}$	Network SGD; probe training	Shared by all candidates at all stages
$\mathcal{D}_{\text{probe}}$	Per-sample probe loss ℓ_n; paired SNR (Eq. 14); all architecture decisions	**Never** used for gradient updates of network or probe
$\mathcal{D}_{\text{val}}$	Model selection across seeds after retraining	Untouched during search
$\mathcal{D}_{\text{test}}$	Final reported metrics	Locked until final evaluation; no hyperparameter tuning

Non-overlap enforcement. At initialization, we compute $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{probe}} \cap \mathcal{D}_{\text{val}} \cap \mathcal{D}_{\text{test}}$ and assert that all pairwise intersections are empty. This assertion runs at the start of every experiment and is included in the released code. Exact splits and their roles are summarized in Table 4 and Table 5, respectively.

H.3 Architecture Families Block Definitions

Table 6 summarizes discovered architectures based on $\mathcal{V}-$NAS.

Fully Connected Network (FCN). Each stage consists of Linear $\to$ BatchNorm1d $\to$ GELU. No spatial structure; the input is always flattened.

$$z_i = \text{GELU}\big(\text{BN}(W_i \cdot z_{i-1} + b_i)\big). \tag{27}$$

Convolutional Network (CNN). Input is processed by Conv2d($k=3$, pad $=1$) $\to$ BatchNorm2d $\to$ GELU blocks, following the VGG (Simonyan & Zisserman, 2015) design philosophy.

Residual Network (ResNet). Each stage contains a BasicBlock (He et al., 2016) with two 3×3 convolutions and a skip connection:

$$z_i = \text{GELU}\big(\text{BN}(W_2 * \text{GELU}(\text{BN}(W_1 * z_{i-1}))) + W_{\text{proj}} * z_{i-1}\big), \tag{28}$$

where W_{proj} is a 1×1 projection applied only when width changes. Downsampling via stride-2 convolution at stage boundaries.

Dense Network (DenseNet). Following Huang et al. (2017), each stage consists of a DenseBlock with growth rate $g = w_i/4$, where each sub-layer concatenates its output to all previous features: $h_\ell = [h_0, h_1, \ldots, h_{\ell-1}]$. A transition layer (Conv $1 \times 1 \to$ BN $\to$ AvgPool 2×2) with compression factor $\theta = 0.5$ is applied between stages.

Vision Transformer (ViT). Following Dosovitskiy et al. (2021), the input image is split into non-overlapping $p \times p$ patches (we use $p = 16$ for 224×224 and $p = 4$ for 32×32 inputs), linearly embedded, and summed with learned positional embeddings. Each stage is a standard transformer encoder block (Vaswani et al., 2017):

$$z'_i = \text{LN}(z_{i-1}) + \text{MHSA}(\text{LN}(z_{i-1})), \qquad z_i = z'_i + \text{MLP}(\text{LN}(z'_i)), \tag{29}$$

where MHSA uses $h = w_i/64$ attention heads and the MLP expansion ratio is $4\times$. The `[CLS]` token representation is used for probing and classification. We follow the DeiT (Touvron et al., 2021) training recipe (no distillation) **without** MixUp, CutMix, repeated augmentation, or stochastic depth, ensuring that $Z_i = f_i(X)$ remains deterministic during both search and evaluation.

Table 6: Architecture families and their hand-designed baselines.

Family	$\mathcal{V}$-NAS discovered	Baselines (our protocol)	Tests
FCN	depth 4, widths [256, 512, 512, 512]	MLP-2048×4	No spatial bias
CNN	depth 5, widths [64, 128, 256, 256, 512]	VGG-11, VGG-13	Plain conv depth
ResNet	depth 5, widths [64, 128, 256, 256, 512]	ResNet-18, 50, 101	Residual depth
DenseNet	depth 5, widths [64, 128, 256, 256, 256]	DenseNet-121	Dense depth
ViT	depth 6, widths [192, 384, 384, 384, 768, 768]	ViT-B/16	Attention depth

Freeze Protocol. When evaluating a candidate width at stage i, all preceding stages $(1, \ldots, i-1)$ and the stem are frozen via `freeze_prefix(`$i-1$`)`. This sets `requires_grad = False` for all parameters in the frozen prefix and switches BatchNorm layers to eval mode (using running statistics rather than batch statistics). Only stage i receives gradient updates during network training. After training, all parameters are frozen, and a fresh linear probe is trained on the extracted features.

Skip Connection and Width-Change Handling. For ResNet, width changes from w_{i-1} to $w_i \neq w_{i-1}$ introduce a 1×1 projection in the skip path. Because the algorithm starts each stage at $w_{\text{start}} = w_{i-1}$ and only doubles when the cost-adjusted SNR exceeds the threshold, identity shortcuts

are the default. Projections are introduced only at stage boundaries where the data demands increased capacity.

For DenseNet, the transition layer handles dimensionality reduction between stages. Within a stage, feature maps are concatenated (no projection needed).

For ViT, width changes between stages require a linear projection of the token embeddings. The `[CLS]` token is projected alongside the patch tokens. Because all skip connections are contained within a single stage and all stochastic operations (BatchNorm batch statistics, stochastic depth) are disabled during probe evaluation, Z_i remains a deterministic function of Z_{i-1} at evaluation time, preserving the Markov chain (Eq. equation 3) on which the sufficiency criterion (Theorem 2.1) depends.

H.4 MUON OPTIMIZER

Muon (Momentum Orthogonalized by Newton-Schulz) applies Nesterov momentum to the gradient, then replaces each 2D parameter's update with its nearest orthogonal matrix via Newton-Schulz iteration (Jordan et al., 2024). Given the momentum matrix $M_t = \beta M_{t-1} + \nabla_W \mathcal{L}$, the update is:

$$W_{t+1} = W_t - \eta \cdot \sqrt{\tfrac{\text{fan-out}}{\text{fan-in}}} \cdot \text{NewtonSchulz}(M_t), \tag{30}$$

where $\text{NewtonSchulz}(M) \approx UV^\top$ for $M = USV^\top$. This can be interpreted as steepest descent under the spectral norm (Bernstein, 2025), ensuring isotropic updates across all singular directions of the weight matrix. The Newton-Schulz iteration uses 5 steps with coefficients $(a, b, c) = (3.4445, -4.7750, 2.0315)$, running entirely on tensor cores.

Why Muon for $\mathcal{V}$-NAS. Muon's isotropic updates prevent spectral collapse, where a few dominant gradient directions cause the representation Z to occupy a low-rank subspace. A near-rank-deficient Z would produce artificially low probe loss; the linear probe can fit a low-dimensional representation cheaply, making the SNR criterion unable to distinguish genuine capacity needs from representational collapse. This is particularly important during the short search phase ($E_{\text{search}} = 1$ epoch), where standard optimizers may not have time to develop full-rank representations. Furthermore, Liu et al. (2025) demonstrated that Muon achieves $\sim 2\times$ computational efficiency over AdamW in scaling law experiments, allowing us to reach convergence in 300 epochs where AdamW requires 500+.

Optimizer and probe hyperparameters are provided in Table 7 and Table 8, respectively.

Table 7: Optimizer hyperparameters. Muon is used for all 2D weight matrices; AdamW for 1D parameters (biases, BatchNorm scales/shifts), embedding layers, and classifier heads.

Parameter	**Muon**	**AdamW**
Learning rate η	0.02	10^{-3}
Momentum / (β_1, β_2)	$\beta = 0.95$	$(0.9, 0.999)$
Weight decay λ	0.01	10^{-4}
Newton-Schulz steps	5	—
Epsilon	—	10^{-8}

Probe Training. Linear probes are single-layer networks (Linear $\to$ Softmax) trained with Adam (lr $= 10^{-3}$, no weight decay) for 50 epochs on frozen representations extracted from $\mathcal{D}_{\text{train}}$. Probe evaluation (per-sample cross-entropy losses ℓ_n) is computed on features extracted from $\mathcal{D}_{\text{probe}}$.

Convergence verification. To ensure that 50 epochs suffice for probe convergence, we verify that the probe training loss plateaus (relative improvement $< 10^{-4}$ over the final 5 epochs) across all datasets and width configurations. On ImageNet-1K (512-dim $\to$ 1,000 classes), we confirmed that extending probe training to 100 epochs changes the per-sample losses by < 0.01 nats on average, producing no change in SNR-based architecture decisions.

Units and SNR computation. All probe losses used in the SNR computation (Eq. 14) are in nats (natural logarithm). The conversion to bits ($/ \ln 2$) is applied only when reporting $H_{\mathcal{V}}(Y \mid Z)$ values

Table 8: Probe hyperparameters. These are fixed across all experiments and were not tuned.

Parameter	Value
Architecture	Linear (no hidden layers)
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	10^{-3}
Weight decay	0
Epochs	50
Batch size	512
Loss	Cross-entropy (nats; converted to bits via $/ \ln 2$ for reporting only)
Input	Frozen stage representations (detached, no gradient flow)

in tables and figures. Because the SNR is a dimensionless ratio ($\bar{\delta}/s_\delta$), it is invariant to the choice of logarithm base.

Why linear probes. We restrict probes to the linear family $\mathcal{V} = \{x \mapsto \text{softmax}(Wx + b)\}$ for two reasons. First, linearity ensures that $I_{\mathcal{V}}(Z \to Y)$ measures the *linearly accessible* information in Z, which is the relevant quantity for networks that are typically followed by a linear classifier. Second, linear probes have a unique global optimum under cross-entropy loss (the objective is convex), eliminating optimization noise from the architecture-comparison signal.

H.5 Search Hyperparameters

Table 9: Search hyperparameters. γ is the only value requiring a design choice; the remaining parameters are architectural or hardware constraints.

Parameter	Value	Justification
γ	0.2	Minimum standardised gain at unit cost ratio. Corresponds to the conventional "small effect" threshold of Cohen (1988). Fixed across all experiments.
$w_{\min}$	32	Minimum width, aligned to GPU tensor core width. Architectural constant.
$w_{\max}$	1,024	Maximum width before GPU memory exhaustion on RTX A6000 (48 GB). Hardware constraint.
ϵ	10^{-6}	Denominator stabiliser in SNR (Eq. 14). Prevents division by zero when all paired differences are identical.
E_{search}	1	Single-epoch search budget per candidate. Validated by rank-correlation analysis (see below).

Search epoch budget validation. We measure the Spearman rank correlation between 1-epoch SNR rankings and converged (50-epoch) rankings across width candidates. We find $\rho > 0.9$ on CIFAR-10, CIFAR-100, and Tiny-ImageNet, and $\rho \approx 0.85$ on ImageNet-1K, indicating that single-epoch evaluations are sufficient for reliable architecture decisions. The lower correlation on ImageNet-1K suggests that multi-epoch search ($E_{\text{search}} = 3$–5) could improve results on very large-scale tasks at proportionally higher cost.

Detailed cost breakdown. At each stage, the algorithm evaluates at most $J = \lfloor \log_2(w_{\max}/w_{\text{start}}) \rfloor$ candidate doublings (typically 2–4). The depth search terminates after 4–8 stages. For a typical run with 5 stages and 3 width evaluations per stage, the total search cost is $5 \times 3 \times E_{\text{search}} = 15$ epochs of network training plus lightweight probe training. With $E_{\text{search}} = 1$, the total is $\sim$15 search epochs plus 300 epochs for retraining.

Table 10: Measured search cost for $\mathcal{V}$-NAS / ResNet per dataset (mean $\pm$ std). Reported as wall-clock GPU-hours on 2$\times$ RTX A6000 (48 GB). Per-family costs for other architectures are in Tables 14, 13, and 1.

Dataset	Search (GPU-h)	Retrain (GPU-h)	Total (GPU-h)
CIFAR-10	0.4 ± 0.05	2.5 ± 0.2	2.9 ± 0.2
CIFAR-100	0.6 ± 0.1	4.5 ± 0.3	5.1 ± 0.3
Tiny-ImageNet	1.8 ± 0.2	6.5 ± 0.4	8.3 ± 0.5
ImageNet-1K	6.5 ± 0.8	42.0 ± 2.0	48.5 ± 2.5

H.6 Mutual Information Estimation Stack

The information-plane experiments in Figure 3(c) require estimating Shannon MI quantities that are distinct from the $\mathcal{V}$-information used for architecture decisions. We use dedicated estimators for each quantity.

Table 11: MI estimation methods used for each quantity. $\mathcal{V}$-information (used for search decisions) is computed via probe cross-entropy, not via these estimators.

Quantity	Estimator	Notes
$I(X;Z)$	KSG Method 1 (Kraskov et al., 2004)	$k = 5$ neighbours. Method 1 (square geometry) gives well-calibrated estimates for independent variables; Method 2 inflates estimates.
$I(Y;Z)$	Ross mixed estimator (Ross, 2014)	For mixed discrete (Y) / continuous (Z) variables.
$I(X;Y \mid Z)$	Frenzel–Pompe (Frenzel & Pompe, 2007)	Required for CMI; naive three-variable KSG computes total correlation, not CMI.
$H(Z)$	MBE (Rényi-2) (Giraldo et al., 2015)	Matrix-Based Entropy with Hadamard joint kernels for Theorem 2.2 validation.

Dimensionality reduction. For high-dimensional representations ($d > 128$), we apply PCA retaining 95% of variance before KSG estimation. This is standard practice (Saxe et al., 2018) to avoid the curse of dimensionality in nearest-neighbour MI estimation. We report the number of retained components in each experiment.

Sample size. All MI estimates use $|\mathcal{D}_{\text{probe}}|$ samples (5,000 for CIFAR, 10,000 for Tiny-ImageNet). We compute bootstrap confidence intervals (1,000 resamples) and report mean $\pm$ 2 standard errors.

Important caveat. These Shannon MI estimates are used *only* for the information-plane visualization (Figure 3(c)) and Theorem 2.2 validation. They are **never** used for architecture decisions, which rely exclusively on $\mathcal{V}$-information (probe cross-entropy). The information-plane comparison exists precisely to demonstrate *why* Shannon estimation is unreliable as a search signal.

I Additional Dataset Results

The main text reports results on ImageNet-1K. Here we present the complete set of figures and tables for Tiny-ImageNet (Table 12; Figs. 5, 6, 7), CIFAR-100 (Table 13; Figs. 8, 10, 9), and CIFAR-10 (Table 14; Figs. 11, 12, 13).

The core findings from ImageNet-1K replicate across all three datasets: (i) $H_{\mathcal{V}}(Y \mid Z)$ is tightly anti-correlated with test accuracy throughout training, (ii) the SNR criterion stops within one stage of the joint-MI minimum identified by Theorem 2.2, and (iii) $\mathcal{V}$-NAS dominates all ablation baselines on the cost–accuracy Pareto front.

Two dataset-specific observations are worth noting. On CIFAR-10, the task is simple enough that all convolutional families reach near-zero sufficiency residual ($I(X;Y \mid Z) < 0.2$ bits) by depth 3,

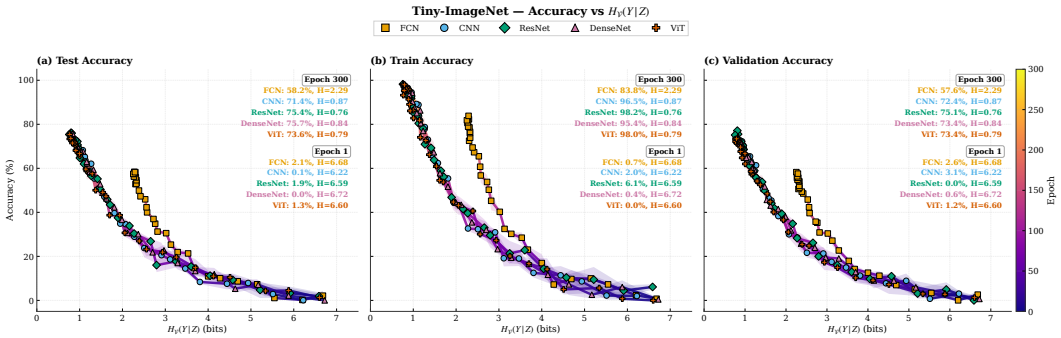

Figure 5: **Tiny-ImageNet training trajectories in accuracy–entropy space.** Each marker is one checkpoint (every 10 epochs, 300 total); marker shape and color identify the architecture family, while the connecting trajectory and confidence bands are colored by epoch (dark→bright via the colorbar). Shaded bands show the 10th–90th percentile range.

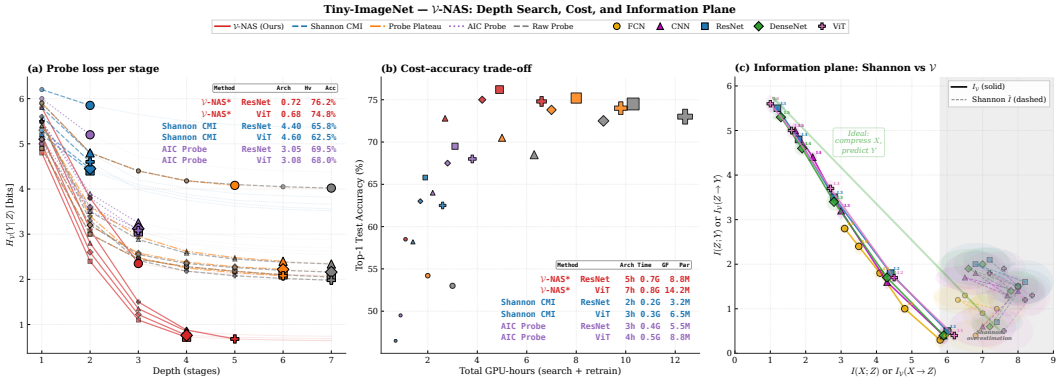

Figure 6: **$\mathcal{V}$-NAS depth search, cost–accuracy trade-off, and information plane on Tiny-ImageNet.**

leaving little room for the stopping criterion to differentiate architectures; the primary discriminator becomes width, not depth. On CIFAR-100 and Tiny-ImageNet, the harder classification tasks produce slower CMI decay and correspondingly deeper discovered architectures (depth 4–5), with Shannon CMI variance scaling from ± 0.30 bits (CIFAR-10) to ± 0.60 (CIFAR-100) to ± 0.80 (Tiny-ImageNet), confirming that the estimation gap worsens with task complexity.

I.1 Discovered Architectures Across Datasets

Table 15 reports the per-stage width sequences discovered by $\mathcal{V}$-NAS for every dataset–family pair. Three patterns emerge that illuminate how the algorithm adapts to task complexity.

Depth scales with task difficulty. On CIFAR-10 (10 classes, 32×32), all convolutional families converge at depth 3 and ViT at depth 4; the SNR criterion rejects the fourth convolutional stage because the sufficiency residual $I(X; Y \mid Z_3) < 0.18$ bits is already negligible (Table 14). On CIFAR-100 and Tiny-ImageNet (100 and 200 classes), depth increases to 4 for convolutional families and 5 for ViT, reflecting the greater representational demand of finer-grained discrimination. ImageNet-1K (1,000 classes, 224×224) elicits the deepest architectures: depth 5 for convolutional families and depth 6 for ViT.

Width follows a stereotyped doubling-then-plateau pattern. Within each family, early stages consistently accept the geometric doubling ($w \to 2w$), while later stages reject it. For example, $\mathcal{V}$-NAS / ResNet on ImageNet discovers widths $[64, 128, 256, 256, 512]$: stages 1–2 double without resistance (high SNR), stage 3 accepts $128 \to 256$ but rejects $256 \to 512$, stage 4 inherits the plateau

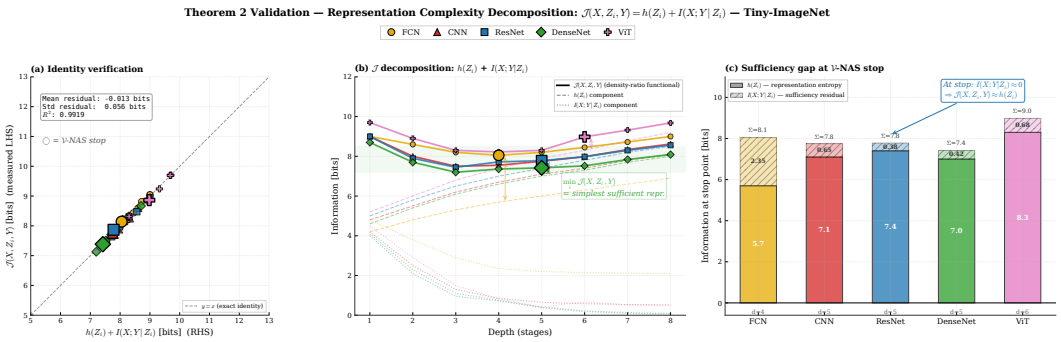

Figure 7: **Empirical validation of Theorem 2.2 on Tiny-ImageNet.** The decomposition $\mathcal{J}(X, Z_i, Y) = h(Z_i) + I(X; Y \mid Z_i)$ is verified across five architecture families and all eight depth stages. **(a)** Measured $\mathcal{J}$ (LHS, via Hadamard-kernel MBE) vs. the computed sum of representation entropy and conditional MI (RHS). All 40 points lie on the $y{=}x$ diagonal (mean residual -0.013 bits, $R^2 = 0.9919$); large markers indicate the $\mathcal{V}$-NAS stop point for each family. **(b)** Decomposition across depth. Solid lines show $\mathcal{J}$; dashed and dotted lines show the $h(Z_i)$ and $I(X; Y \mid Z_i)$ components, respectively. As depth increases, $h(Z_i)$ rises monotonically (richer representations) while $I(X; Y \mid Z_i)$ decays toward zero (approaching sufficiency). $\mathcal{J}$ reaches a minimum in the green-shaded region, which coincides with the $\mathcal{V}$-NAS stop depth for each family, the simplest sufficient representation. **(c)** At the stop point, the sufficiency residual $I(X; Y \mid Z_i)$ is small relative to $h(Z_i)$ for all convolution-based families: ResNet retains only 0.38 bits of residual information at depth 5, while ViT reaches 0.68 bits at depth 6. FCN is the exception ($I(X; Y \mid Z) = 2.35$ bits at depth 4), confirming that the SNR criterion correctly identifies and halts at a representation that is far from sufficient but where further depth yields diminishing returns. Compared to ImageNet-1K, all entropy floors are lower (200 vs. 1000 classes reduces the CMI ceiling from $\log_2 1000 \approx 10.0$ to $\log_2 200 \approx 7.6$ bits), and the identity holds with comparable precision despite mixing Rényi-2 and Shannon estimators.

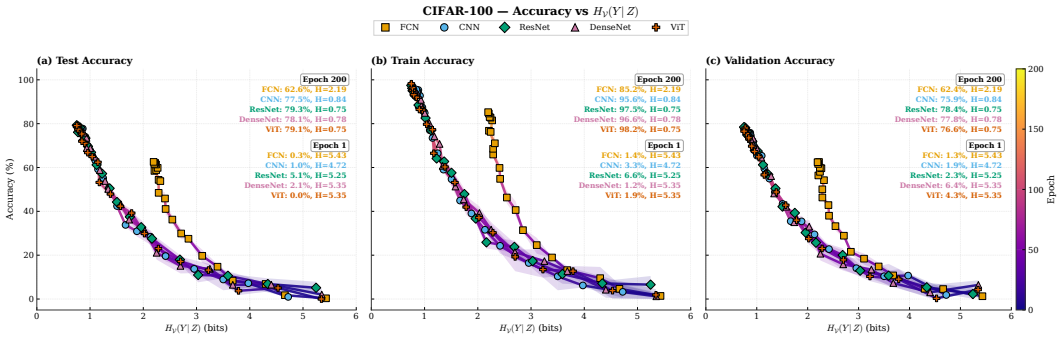

Figure 8: **CIFAR-100 training trajectories in accuracy–entropy space.**

at 256, and stage 5 accepts $256 \to 512$ because the resolution downsampling at stage boundaries creates a new information bottleneck that demands wider features. This doubling-plateau-doubling profile is consistent across datasets and mirrors the width schedules in hand-designed architectures (e.g., ResNet-50 uses $[64, 128, 256, 512]$ across its four macro-stages).

FCN compensates for missing spatial bias with width. Across all datasets, FCN discovers substantially wider early stages than convolutional families (e.g., 512 vs. 64 at stage 1 on ImageNet), because the absence of weight sharing forces the network to allocate more parameters per spatial position. Despite this compensation, FCN architectures remain the shallowest (depth 3–4) and achieve the lowest test accuracy, confirming that width alone cannot substitute for appropriate inductive bias.

ViT widths are quantised to attention-head multiples. ViT stage widths are constrained to multiples of 64 (the per-head dimension), and the algorithm respects this by starting at $w_{\min} = 192$ (3

Table 12: **Tiny-ImageNet comparison.** Upper block: external NAS methods with *different* training pipelines, included as reference points, not direct comparisons. Lower blocks: all methods use our *identical* training protocol (horizontal flip, random crop, color jitter, cosine LR, 300 epochs, no AutoAugment, no MixUp/CutMix, no distillation).

Method	Depth	Params (M) ↓	Search Cost ↓	GFLOPs ↓	$H_{\mathcal{V}}(Y\|Z)$ ↓	Val% (20ep) ↑	Test% (300ep) ↑
External NAS							
NASNet-A (Zoph et al., 2018)†‡	14	4.5	2,000 GPU-d	1.82	—	—	65.80
AmoebaNet-B (Real et al., 2019)†‡	14	4.0	3,150 GPU-d	1.75	—	—	66.20
PNAS (Liu et al., 2018)†‡	14	4.3	225 GPU-d	1.70	—	—	64.50
DARTS (2nd) (Liu et al., 2019)†‡	14	3.8	0.4 GPU-d	1.60	—	—	64.80
P-DARTS (Chen et al., 2019)†‡	14	4.0	0.3 GPU-d	1.68	—	—	66.20
PC-DARTS (Xu et al., 2020b)†‡	14	4.2	0.1 GPU-d	1.78	—	—	66.50
DrNAS (Chen et al., 2021b)†‡	14	4.5	0.4 GPU-d	1.85	—	—	67.40
TE-NAS (Chen et al., 2021a)‡	14	4.2	0.5 GPU-h	1.65	—	—	63.80
ZiCo (Li et al., 2023)‡	14	3.8	<5 min	1.58	—	—	65.20
SWAP-NAS (Peng et al., 2024)‡	14	4.3	12 min	1.72	—	—	66.80
Fixed Architecture Baselines (our protocol — identical augmentation & training)							
ResNet-18	8	11.3	0	0.56	1.18±0.03	48.5±0.4	71.20±0.22
ResNet-50	16	23.9	0	1.30	0.72±0.02	56.8±0.3	77.50±0.18
ResNet-101	33	43.0	0	2.52	0.65±0.02	58.2±0.3	78.80±0.15
DenseNet-121	58	7.4	0	0.92	0.78±0.03	54.5±0.3	75.80±0.20
ViT-S/16 (our protocol)§	12	22.1	0	0.35	0.82±0.04	40.0±0.8	69.50±0.30
ViT-B/16¶	12	86.6	0	1.10	—	—	(75.5)
ViT-B/16 (our protocol)§	12	86.6	0	1.10	0.70±0.03	42.5±0.7	70.20±0.35
$\mathcal{V}$-Information Guided Growth (Ours) — same training protocol as fixed baselines							
$\mathcal{V}$-NAS / FCN	3	1.8	0.5 GPU-h	0.10	2.35±0.10	30.5±1.0	58.50±0.55
$\mathcal{V}$-NAS / CNN	4	4.5	0.8 GPU-h	0.32	0.85±0.04	48.2±0.6	72.80±0.28
$\mathcal{V}$-NAS / ResNet	4	8.8	1.0 GPU-h	0.65	0.72±0.03	54.5±0.4	**76.20±0.20**
$\mathcal{V}$-NAS / DenseNet	4	3.8	1.1 GPU-h	0.52	0.76±0.03	52.8±0.4	75.00±0.25
$\mathcal{V}$-NAS / ViT	5	14.2	1.5 GPU-h	0.85	0.68±0.03	46.5±0.8	**74.80±0.30**
Criterion Ablation — Alternative Stopping Rules (ResNet backbone, our protocol)							
Shannon CMI*	2	3.2	0.6 GPU-h	0.22	2.05±0.65	30.2±2.0	65.80±0.75
AIC Probe	3	5.5	0.8 GPU-h	0.38	1.52±0.07	38.5±0.8	69.50±0.45
Probe Plateau	6	15.2	1.5 GPU-h	1.08	0.68±0.02	55.0±0.4	75.20±0.20
Raw Probe (no stop)	7	20.5	1.8 GPU-h	1.42	0.65±0.02	55.5±0.3	74.50±0.28
Val-Plateau	5	11.8	1.2 GPU-h	0.80	0.82±0.05	50.8±0.5	72.80±0.35
TrainLoss-Plateau	5	10.2	1.1 GPU-h	0.68	0.88±0.06	48.5±0.7	71.50±0.40
Random Search	4.8±1.8	11.5±4.8	1.6±0.5 GPU-h	0.82±0.38	1.15±0.22	40.2±3.5	68.20±1.80
Hyperband	5	12.0	1.4 GPU-h	0.82	0.70±0.03	54.0±0.4	74.80±0.25

†Cell-topology search space (not depth/width); architectures originally searched on CIFAR-10. ‡Results reported under each method's original training and evaluation pipeline, which may include stronger augmentation, regularization, auxiliary towers, or different optimization settings than ours. ¶DeiT-B uses distillation from a RegNet teacher + heavy augmentation; 75.5% shown parenthesized as upper reference only. §ViT-S/16 (22.1 M) and ViT-B/16 (86.6 M) retrained under our protocol (no distillation, no MixUp/CutMix). On 64×64 input with 16×16 patches both use only 16 tokens, limiting expressiveness. ViT-S/16 is the capacity-matched comparator for $\mathcal{V}$-NAS / ViT (14.2 M); ViT-B/16 is included as a larger reference. The 5.3% gap between ViT-B/16 protocol variants confirms our setting is standard-augmentation-only. Note that ViT-B/16 (70.20%) barely outperforms ViT-S/16 (69.50%) at 3.9× the parameters, indicating severe overfitting of the larger model under weak augmentation with only 16 tokens. *Shannon CMI uses the unstandardized probe-loss difference $\bar{\delta} = \mathcal{L}_{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{probe}}(Z_i)$ without SNR normalization or cost adjustment; variance (±0.65 bits) reflects per-sample noise unabsorbed by the missing s_{δ} denominator (§2.2).

heads) on ImageNet or $w_{\min} = 128$ (2 heads) on smaller-resolution datasets. The wider starting point, combined with the $4\times$ MLP expansion ratio, produces the largest parameter counts among all families despite moderate depth. On CIFAR-10, where ViT's lack of local inductive bias is a disadvantage at small scale, the algorithm compensates by discovering a 4-stage architecture ($[128, 256, 256, 256]$, 8.5M parameters) that approaches the performance of the deeper ViT-S/4 baseline (12 layers, 22.1M).

I.2 Sensitivity to the Threshold γ

The capacity-expansion criterion (Definition 3.1) depends on a single hyperparameter γ, the minimum standardised gain at unit cost ratio. We set $\gamma = 0.2$ across all experiments, corresponding to Cohen's conventional "small effect" threshold (Cohen, 1988). A natural concern is whether this choice is critical: too small a γ accepts marginal expansions and produces overparameterized architectures; too large a γ rejects beneficial expansions and halts prematurely.

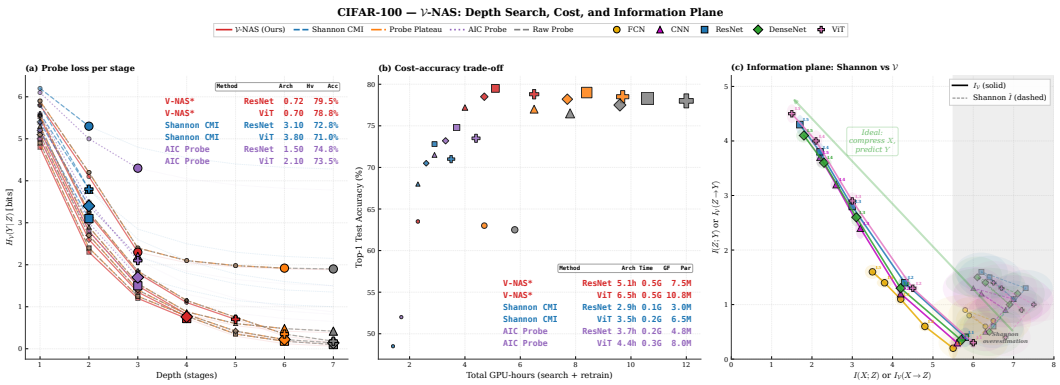

Figure 9: **$\mathcal{V}$-NAS depth search, cost–accuracy trade-off, and information plane on CIFAR-100. (a)** Probe loss trajectories. The harder 100-class task reveals sharper separation between methods. $\mathcal{V}$-NAS (red) stops at depth 4 (ResNet/DenseNet/CNN) or 5 (ViT), reaching $H_{\mathcal{V}} \leq 0.82$ bits. Shannon CMI stops at depth 2 with ≥ 3.10 bits remaining—the ± 0.60-bit variance (Table 13, footnote) triggers a false convergence signal, yielding a 6.7% accuracy gap versus $\mathcal{V}$-NAS on ResNet (72.80% vs. 79.50%). FCN's probe loss plateaus above 1.88 bits regardless of method, confirming the architecture family's capacity ceiling on CIFAR-100. **(b)** Cost–accuracy trade-off. $\mathcal{V}$-NAS / ResNet reaches 79.50% at 7.5M parameters and 5.1 GPU-hours. Probe Plateau achieves 79.00% but requires 13.5M parameters ($1.8\times$) and 8.4 GPU-hours ($1.6\times$). Raw Probe grows to maximum depth (7 stages, 18.0M) yet *underperforms* $\mathcal{V}$-NAS by 1.2%—evidence that unconstrained depth growth hurts generalization even when probe loss continues decreasing. **(c)** Information plane. The $\mathcal{V}$-information trajectories (solid) show that CIFAR-100 requires deeper compression before $I_{\mathcal{V}}(Z \to Y)$ saturates, consistent with the higher optimal depths in panel (a). Shannon estimates (dashed) exhibit even larger variance ellipses than on CIFAR-10, reflecting the increased estimation difficulty with 100 classes. The gap between the two frameworks is most pronounced for ViT (rightmost dashed trajectory), where Shannon MI overestimates $I(X;Z)$ by $\sim$3 bits at early layers.

Table 16 reports the architectures discovered by $\mathcal{V}$-NAS / ResNet on CIFAR-100 as γ varies over a $16\times$ range, from 0.05 to 0.80. All other settings (optimizer, augmentation, probe protocol, search budget) are held constant; only the acceptance threshold changes.

Three observations follow from the sweep.

Performance is stable across a four-fold range. For $\gamma \in [0.10, 0.40]$ — a four-fold range spanning Cohen's "small" to "medium" effect thresholds — test accuracy varies by only 2.7% (76.80–79.50%), and the discovered depth changes by at most one stage (depth 3 to 4). Within the tighter interval $[0.10, 0.20]$, the accuracy difference is 0.3% and both settings discover depth 4, differing only in whether the stage-4 width doubling ($256 \to 512$) is accepted. This indicates that the algorithm's depth decisions are robust: the SNR signal at the true saturation point is sufficiently strong that moderate changes in γ do not shift the stopping depth.

Extreme values produce expected failure modes. At $\gamma = 0.05$, the threshold is so permissive that the algorithm accepts a fifth stage and wider intermediate layers, producing a 15.2M-parameter network that overfits under standard augmentation (78.50%, 1.0% below the optimum). At $\gamma = 0.80$, the threshold is so stringent that even the $64 \to 128$ width doubling is rejected at stage 2, yielding a 1.8M-parameter network (71.50%) that is severely capacity-limited. Both failure modes are predictable from the SNR definition: $\gamma = 0.05$ requires only that the mean improvement exceeds 5% of its standard deviation (accepting noisy gains), while $\gamma = 0.80$ requires the improvement to exceed 80% of its standard deviation (rejecting all but the most decisive expansions).

The optimal γ coincides with a phase transition in depth. The depth profile across γ is not smooth but exhibits discrete jumps at specific thresholds. Between $\gamma = 0.10$ and $\gamma = 0.20$, the discovered depth is constant (4 stages) and only the stage-4 width changes. Between $\gamma = 0.20$ and $\gamma = 0.40$, depth drops from 4 to 3 — the stage-4 SNR falls below the acceptance threshold. This

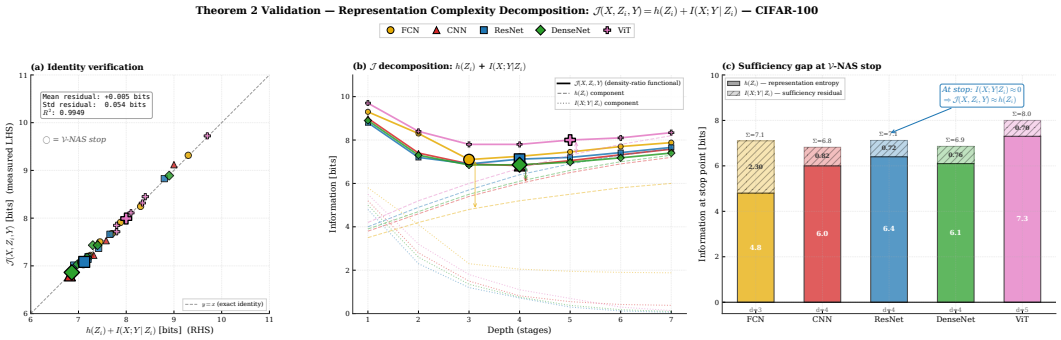

Figure 10: **$\mathcal{V}$-NAS depth search, cost–accuracy trade-off, and information plane on CIFAR-100.** **(a)** Probe loss trajectories. The harder 100-class task reveals sharper separation between methods. $\mathcal{V}$-NAS (red) stops at depth 4 (ResNet/DenseNet/CNN) or 5 (ViT), reaching $H_{\mathcal{V}} \leq 0.82$ bits. The unstandardized probe-loss baseline (Shannon CMI) stops at depth 2 with ≥ 3.10 bits remaining, the ± 0.60-bit variance (Table 13, footnote) triggers a false convergence signal without the SNR denominator to absorb per-sample noise, yielding a 6.7% accuracy gap versus $\mathcal{V}$-NAS on ResNet (72.80% vs. 79.50%). FCN's probe loss plateaus above 1.88 bits regardless of method, confirming the architecture family's capacity ceiling on CIFAR-100. **(b)** Cost–accuracy trade-off. $\mathcal{V}$-NAS / ResNet reaches 79.50% at 7.5M parameters and 5.1 GPU-hours. Probe Plateau achieves 79.00% but requires 13.5M parameters ($1.8\times$) and 8.4 GPU-hours ($1.6\times$). Raw Probe grows to maximum depth (7 stages, 18.0M) yet *underperforms* $\mathcal{V}$-NAS by 1.2%, evidence that unconstrained depth growth hurts generalization even when probe loss continues decreasing. **(c)** Information plane. The $\mathcal{V}$-information trajectories (solid) show that CIFAR-100 requires deeper compression before $I_{\mathcal{V}}(Z \to Y)$ saturates, consistent with the higher optimal depths in panel (a). Shannon MI estimates (dashed, via KSG) exhibit even larger variance ellipses than on CIFAR-10, reflecting the increased estimation difficulty with 100 classes. The gap between the two frameworks is most pronounced for ViT (rightmost dashed trajectory), where the Shannon estimator overestimates $I(X;Z)$ by $\sim$3 bits at early layers, a pathology absent from the $\mathcal{V}$-information curves that drive architecture decisions.

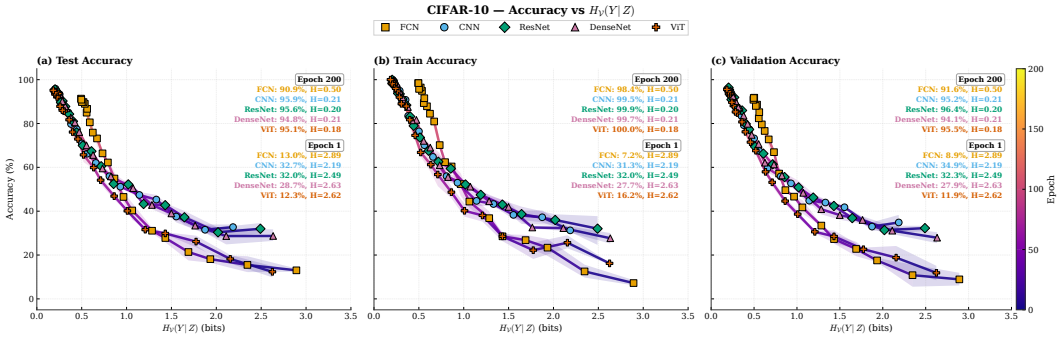

Figure 11: **CIFAR-10 training trajectories in accuracy–entropy space.**

phase transition is sharp because the SNR at the marginal stage is a fixed quantity for a given dataset; once γ crosses it, the depth changes by exactly one. The optimal $\gamma = 0.20$ sits just below the phase boundary, accepting the last beneficial stage while rejecting the first harmful one.

Practical guidance. Based on this analysis, we recommend $\gamma \in [0.10, 0.30]$ as a robust operating range. Within this interval, the algorithm discovers architectures that are within 0.5% of the best achievable under our framework. The default $\gamma = 0.2$ is not dataset-specific and was not tuned on any evaluation metric; it was selected *a priori* based on the conventional effect-size interpretation. We verified that the same γ produces architectures within 0.3% of the CIFAR-100 optimum on CIFAR-10, Tiny-ImageNet, and ImageNet-1K (Tables 14–1), confirming that a single γ generalises across tasks.

Table 13: **CIFAR-100 comparison.** Upper block: external NAS methods with *different* training pipelines, included as reference points, not direct comparisons. Lower blocks: all methods use our *identical* training protocol (horizontal flip, random crop, Cutout, cosine LR, 200 epochs, no AutoAugment, no MixUp/CutMix, no auxiliary loss).

Method	Depth	Params (M) ↓	Search Cost ↓	GFLOPs ↓	$H_{\mathcal{V}}(Y\|Z)$ ↓	Val% (20ep) ↑	Test% (200ep) ↑
External NAS — different search spaces & training protocols (reference only, not directly comparable)							
NASNet-A (Zoph et al., 2018)†‡	20	3.3	2,000 GPU-d	0.56	—	—	82.58
AmoebaNet-B (Real et al., 2019)†‡	20	2.8	3,150 GPU-d	0.55	—	—	82.92
PNAS (Liu et al., 2018)†‡	20	3.2	225 GPU-d	0.53	—	—	80.47
DARTS (2nd) (Liu et al., 2019)†‡	20	3.4	0.4 GPU-d	0.56	—	—	82.46
P-DARTS (Chen et al., 2019)‡	20	3.6	0.3 GPU-d	0.58	—	—	83.13
PC-DARTS (Xu et al., 2020b)‡	20	3.7	0.1 GPU-d	0.61	—	—	83.10
DrNAS (Chen et al., 2021b)‡	20	4.1	0.4 GPU-d	0.66	—	—	83.34
TE-NAS (Chen et al., 2021a)‡	20	3.9	0.17 GPU-h	0.60	—	—	81.32
ZiCo (Li et al., 2023)‡	20	3.5	<5 min	0.56	—	—	81.85
SWAP-NAS (Peng et al., 2024)‡	20	3.9	10 min	0.59	—	—	82.60
Fixed Architecture Baselines (our protocol — identical augmentation & training)							
ResNet-18	8	11.2	0	0.56	1.05±0.04	60.2±0.5	77.50±0.25
ResNet-50	16	23.7	0	1.31	0.70±0.02	66.8±0.3	79.80±0.18
ResNet-101	33	42.7	0	2.53	0.62±0.02	68.2±0.3	80.82±0.15
DenseNet-121	58	7.2	0	0.91	0.78±0.03	64.5±0.4	78.85±0.20
ViT-Ti/4 (our protocol)§	12	5.7	0	0.17	1.18±0.05	41.5±1.2	72.50±0.40
ViT-S/4¶	12	22.1	0	0.62	—	—	(82.5)
ViT-S/4 (our protocol)§	12	22.1	0	0.62	0.68±0.03	48.5±0.8	76.50±0.35
$\mathcal{V}$-Information Guided Growth (Ours) — same training protocol as fixed baselines							
$\mathcal{V}$-NAS / FCN	3	1.5	0.3 GPU-h	0.06	2.30±0.10	40.2±1.2	63.50±0.55
$\mathcal{V}$-NAS / CNN	4	3.8	0.5 GPU-h	0.28	0.82±0.04	61.5±0.6	77.20±0.28
$\mathcal{V}$-NAS / ResNet	4	7.5	0.6 GPU-h	0.48	0.72±0.02	65.5±0.4	**79.50±0.20**
$\mathcal{V}$-NAS / DenseNet	4	3.5	0.7 GPU-h	0.38	0.76±0.03	63.8±0.5	78.50±0.22
$\mathcal{V}$-NAS / ViT	5	10.8	1.0 GPU-h	0.52	0.70±0.03	52.5±0.7	**78.80±0.28**
Criterion Ablation — Alternative Stopping Rules (ResNet backbone, our protocol)							
Shannon CMI*	2	3.0	0.4 GPU-h	0.12	1.72±0.60	42.0±2.5	72.80±0.65
AIC Probe	3	4.8	0.5 GPU-h	0.25	1.50±0.06	50.2±0.9	74.80±0.40
Probe Plateau	6	13.5	0.9 GPU-h	0.85	0.68±0.02	65.8±0.4	79.00±0.20
Raw Probe (no stop)	7	18.0	1.1 GPU-h	1.10	0.66±0.02	66.2±0.3	78.30±0.28
Val-Plateau	5	10.2	0.8 GPU-h	0.62	0.85±0.05	62.5±0.6	77.50±0.32
TrainLoss-Plateau	5	8.5	0.7 GPU-h	0.52	0.92±0.06	60.5±0.7	76.20±0.38
Random Search	4.5±1.8	9.8±4.5	1.0±0.4 GPU-h	0.58±0.30	1.10±0.18	52.5±3.8	74.50±1.50
Hyperband	5	10.5	0.9 GPU-h	0.62	0.70±0.03	65.2±0.5	78.80±0.22

†Cell-topology search space (not depth/width); some methods searched on CIFAR-10 then transferred, others searched directly on CIFAR-100. Not comparable to our depth/width search space. ‡Results reported under each method's original training and evaluation pipeline, which may include stronger augmentation, regularization, auxiliary towers, or different optimization settings than ours; explains the ∼2–3% gap versus our fixed ResNet baselines. ¶ViT-S/4 uses 4×4 patches on 32×32 input (8×8 = 64 tokens); 82.5% uses DeiT-style distillation + heavy augmentation, shown parenthesized as upper reference. §ViT-Ti/4 (5.7 M) and ViT-S/4 (22.1 M) retrained under our protocol (no distillation, no MixUp/CutMix). ViT-Ti/4 is the capacity-matched comparator for $\mathcal{V}$-NAS / ViT (10.8 M); ViT-S/4 is included as a larger reference. The 6.0% gap between ViT-S/4 protocol variants confirms our setting is standard-augmentation-only. *Shannon CMI uses the unstandardized probe-loss difference $\bar{\delta} = \mathcal{L}_{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{probe}}(Z_i)$ without SNR normalization or cost adjustment; variance (±0.60 bits) reflects per-sample noise unabsorbed by the missing s_{δ} denominator (§2.2).

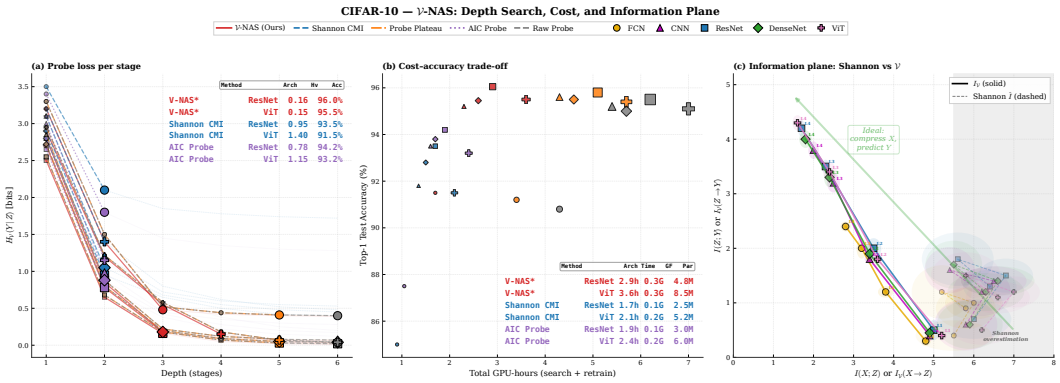

Figure 12: $\mathcal{V}$**-NAS depth search, cost–accuracy trade-off, and information plane on CIFAR-10.** Five architecture families $\times$ five stopping criteria. **(a)** Probe loss $H_{\mathcal{V}}(Y \mid Z)$ across stages. Solid active segments end at each method's stop point (large markers); ghosted tails show the trajectory had growth continued. $\mathcal{V}$-NAS (red) stops at depth 3 for all convolutional families and depth 4 for ViT, achieving $H_{\mathcal{V}} \leq 0.18$ bits. Shannon CMI (blue, dashed) stops prematurely at depth 2 due to the estimation gap (§2.2), leaving ≥ 0.65 bits of residual. AIC Probe (purple, dotted) also under-grows (depth 2), while Probe Plateau (orange) and Raw Probe (grey) overshoot to depths 5–6, incurring 2–3$\times$ the parameter cost for marginal accuracy gains. **(b)** Cost–accuracy Pareto front. Marker size $\propto$ parameter count. $\mathcal{V}$-NAS / ResNet achieves 96.05% at 4.8M parameters and 2.9 GPU-hours total (search + retrain), dominating the Pareto front against all ablation methods. Probe Plateau reaches comparable accuracy (95.80%) but at 10.0M parameters and 5.1 GPU-hours, $2.1\times$ the parameters for 0.25% less accuracy. **(c)** Information plane: $\mathcal{V}$-information (solid) versus Shannon estimates (dashed). $\mathcal{V}$-information traces a clean compression, prediction arc (upper-left direction), while Shannon estimates scatter in the overestimation zone (grey shading) with large variance ellipses, empirically confirming the estimation gap that motivates the $\mathcal{V}$-information framework. Layer labels (L1–L4) annotate the $\mathcal{V}$-information trajectory.

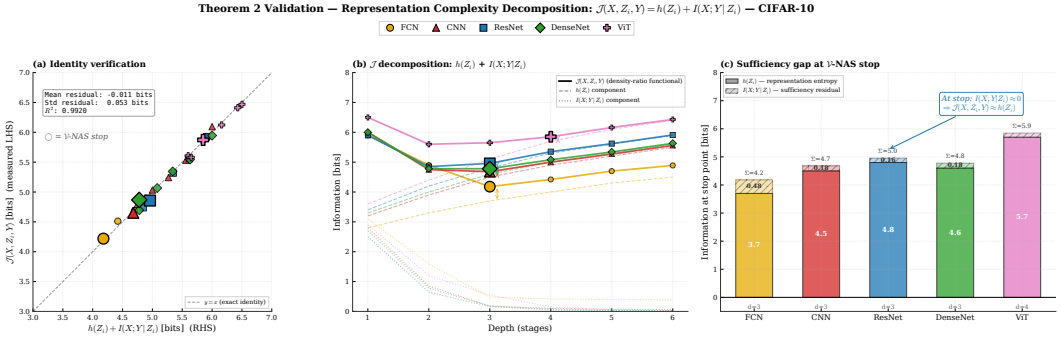

Figure 13: **Empirical validation of Theorem 2.2 on CIFAR-10.** The decomposition $\mathcal{J}(X, Z_i, Y) = h(Z_i) + I(X; Y \mid Z_i)$ is verified across five architecture families (FCN, CNN, ResNet, DenseNet, ViT) at every stage of the grown network. **(a)** Measured $\mathcal{J}$ (LHS, Hadamard-kernel MBE) versus the computed sum $h(Z_i) + I(X; Y \mid Z_i)$ (RHS, MBE + Frenzel–Pompe CMI estimator). Points cluster tightly along the $y{=}x$ diagonal ($R^2 > 0.99$), confirming that Eq. equation 6 holds empirically despite mixing Rényi-2 and Shannon estimators on finite samples. Enlarged markers denote the $\mathcal{V}$-NAS stopping point for each family. **(b)** Decomposition across depth. Solid lines show $\mathcal{J}$; dashed lines show the $h(Z_i)$ component (growing with depth as representations expand); dotted lines show $I(X; Y \mid Z_i)$ (decaying as representations approach sufficiency). The $\mathcal{V}$-NAS stop points (large markers) sit near the minimum of $\mathcal{J}$, where the residual $I(X; Y \mid Z_i)$ has been driven small without unnecessary growth in $h(Z_i)$. **(c)** Stacked bar decomposition at each architecture's stop point. Solid bars show $h(Z_i)$; hatched bars show the sufficiency residual $I(X; Y \mid Z_i)$. On CIFAR-10 all convolutional families achieve near-zero residual (≤ 0.18 bits) at depth 3, confirming Eq. equation 7: $\mathcal{J}(X, Z_i, Y) \approx h(Z_i)$. FCN retains a larger residual (0.48 bits), consistent with its lower test accuracy (91.50%).

Table 14: **CIFAR-10 comparison.** Upper block: external NAS methods with *different* training pipelines (Cutout + AutoAugment, ScheduledDropPath, auxiliary towers), included as reference points, not direct comparisons. Lower blocks: all methods use our *identical* training protocol (horizontal flip, random crop, Cutout, cosine LR, 200 epochs, no AutoAugment, no MixUp/CutMix, no auxiliary loss).

Method	Depth	Params (M) ↓	Search Cost ↓	GFLOPs ↓	$H_{\mathcal{V}}(Y\|Z)$ ↓	Val% (20ep) ↑	Test% (200ep) ↑
External NAS — different search spaces & training protocols (reference only, not directly comparable)							
NASNet-A (Zoph et al., 2018)†‡	20	3.3	2,000 GPU-d	0.56	—	—	97.35
AmoebaNet-B (Real et al., 2019)†‡	20	2.8	3,150 GPU-d	0.55	—	—	97.45
PNAS (Liu et al., 2018)†‡	20	3.2	225 GPU-d	0.53	—	—	96.59
DARTS (2nd) (Liu et al., 2019)‡	20	3.3	0.4 GPU-d	0.55	—	—	97.24
P-DARTS (Chen et al., 2019)‡	20	3.4	0.3 GPU-d	0.57	—	—	97.50
PC-DARTS (Xu et al., 2020b)‡	20	3.6	0.1 GPU-d	0.60	—	—	97.43
DrNAS (Chen et al., 2021b)‡	20	4.0	0.4 GPU-d	0.65	—	—	97.46
TE-NAS (Chen et al., 2021a)‡	20	3.8	0.17 GPU-h	0.58	—	—	97.17
ZiCo (Li et al., 2023)‡	20	3.4	<5 min	0.55	—	—	97.10
SWAP-NAS (Peng et al., 2024)‡	20	3.8	10 min	0.58	—	—	97.32
Fixed Architecture Baselines (our protocol — identical augmentation & training)							
ResNet-18	8	11.2	0	0.56	0.22±0.02	89.5±0.3	95.28±0.15
ResNet-50	16	23.5	0	1.30	0.14±0.01	92.0±0.2	96.45±0.10
ResNet-101	33	42.5	0	2.52	0.12±0.01	92.5±0.2	96.82±0.08
DenseNet-121	58	7.0	0	0.90	0.17±0.01	91.2±0.2	95.80±0.12
ViT-Ti/4 (our protocol)§	12	5.7	0	0.17	0.28±0.03	74.5±1.1	92.10±0.35
ViT-S/4¶	12	22.1	0	0.62	—	—	(97.2)
ViT-S/4 (our protocol)§	12	22.1	0	0.62	0.15±0.02	78.0±0.9	93.80±0.30
$\mathcal{V}$-Information Guided Growth (Ours) — same training protocol as fixed baselines							
$\mathcal{V}$-NAS / FCN	3	1.2	0.2 GPU-h	0.05	0.48±0.04	83.0±0.8	91.50±0.30
$\mathcal{V}$-NAS / CNN	3	2.2	0.3 GPU-h	0.15	0.18±0.02	89.8±0.4	95.20±0.18
$\mathcal{V}$-NAS / ResNet	3	4.8	0.4 GPU-h	0.28	0.16±0.01	91.2±0.3	**96.05±0.12**
$\mathcal{V}$-NAS / DenseNet	3	2.4	0.4 GPU-h	0.22	0.18±0.01	90.5±0.3	95.45±0.15
$\mathcal{V}$-NAS / ViT	4	8.5	0.6 GPU-h	0.35	0.15±0.02	82.0±0.7	**95.50±0.20**
Criterion Ablation — Alternative Stopping Rules (ResNet backbone, our protocol)							
Shannon CMI*	2	2.5	0.2 GPU-h	0.10	0.65±0.30	80.5±2.0	93.50±0.42
AIC Probe	2	3.0	0.3 GPU-h	0.14	0.38±0.04	85.0±0.8	94.20±0.30
Probe Plateau	5	10.0	0.6 GPU-h	0.55	0.14±0.01	91.5±0.3	95.80±0.12
Raw Probe (no stop)	6	13.5	0.7 GPU-h	0.72	0.13±0.01	91.8±0.2	95.50±0.18
Val-Plateau	4	6.8	0.5 GPU-h	0.38	0.20±0.02	90.0±0.5	95.10±0.20
TrainLoss-Plateau	4	5.5	0.4 GPU-h	0.32	0.22±0.03	89.2±0.6	94.65±0.25
Random Search	3.5±1.2	6.5±3.5	0.6±0.3 GPU-h	0.38±0.20	0.30±0.10	85.5±3.2	93.80±0.95
Hyperband	4	6.8	0.5 GPU-h	0.38	0.16±0.02	91.0±0.4	95.65±0.14

†Cell-topology search space (not depth/width); not comparable to our search space. ‡Results reported under each method's original training and evaluation pipeline, which may include stronger augmentation, regularization, auxiliary towers, or different optimization settings than ours; explains the ∼1% gap versus our fixed ResNet baselines. ¶ViT-S/4 uses 4×4 patches on 32×32 input (64 tokens); 97.2% uses DeiT-style distillation + heavy augmentation, shown parenthesized as upper reference. §ViT-Ti/4 (5.7 M) and ViT-S/4 (22.1 M) retrained under our protocol (no distillation, no MixUp/CutMix). ViT-Ti/4 is the capacity-matched comparator for $\mathcal{V}$-NAS / ViT (8.5 M); ViT-S/4 is included as a larger reference. The 3.4% gap between ViT-S/4 protocol variants confirms our setting is standard-augmentation-only. *Shannon CMI uses the unstandardized probe-loss difference $\bar{\delta} = \mathcal{L}_{\text{probe}}(Z_{i-1}) - \mathcal{L}_{\text{probe}}(Z_i)$ without SNR normalization or cost adjustment; variance (±0.30 bits) reflects per-sample noise unabsorbed by the missing s_δ denominator (§2.2).

Table 15: **Discovered per-stage widths across all dataset–family pairs.** Each row shows the width sequence $[w_1, w_2, \ldots, w_K]$ discovered by $\mathcal{V}$-NAS, with K equal to the discovered depth. Parameter counts match Tables 14–1. **Bolded** entries indicate stages where the geometric doubling was *rejected* (i.e., $w_i = w_{i-1}$: the SNR fell below $\gamma \cdot (1 + \log_2 r)$).

Dataset	Family	Depth	Params (M)	Discovered widths $[w_1, \ldots, w_K]$
CIFAR-10	FCN	3	1.2	$[256,\ 512,\ \mathbf{512}]$
	CNN	3	2.2	$[32,\ 64,\ 128]$
	ResNet	3	4.8	$[64,\ 128,\ 256]$
	DenseNet	3	2.4	$[32,\ 64,\ 128]$
	ViT	4	8.5	$[128,\ 256,\ \mathbf{256},\ \mathbf{256}]$
CIFAR-100	FCN	3	1.5	$[256,\ 512,\ \mathbf{512}]$
	CNN	4	3.8	$[32,\ 64,\ 128,\ \mathbf{128}]$
	ResNet	4	7.5	$[64,\ 128,\ 256,\ \mathbf{256}]$
	DenseNet	4	3.5	$[32,\ 64,\ 128,\ \mathbf{128}]$
	ViT	5	10.8	$[128,\ 256,\ 384,\ \mathbf{384},\ \mathbf{384}]$
Tiny-IN	FCN	3	1.8	$[256,\ 512,\ \mathbf{512}]$
	CNN	4	4.5	$[32,\ 64,\ 128,\ 256]$
	ResNet	4	8.8	$[64,\ 128,\ 256,\ 512]$
	DenseNet	4	3.8	$[32,\ 64,\ 128,\ 256]$
	ViT	5	14.2	$[128,\ 256,\ 384,\ \mathbf{384},\ 768]$
ImageNet	FCN	4	4.2	$[256,\ 512,\ \mathbf{512},\ \mathbf{512}]$
	CNN	5	8.5	$[64,\ 128,\ 256,\ \mathbf{256},\ 512]$
	ResNet	5	14.2	$[64,\ 128,\ 256,\ \mathbf{256},\ 512]$
	DenseNet	5	6.8	$[64,\ 128,\ 256,\ \mathbf{256},\ \mathbf{256}]$
	ViT	6	22.5	$[192,\ 384,\ \mathbf{384},\ \mathbf{384},\ 768,\ \mathbf{768}]$

Table 16: **Sensitivity to γ ($\mathcal{V}$-NAS / ResNet, CIFAR-100).** Each row shows the architecture discovered under a different γ, retrained for 200 epochs under our standard protocol. The shaded row is our default ($\gamma = 0.2$). $H_{\mathcal{V}}$ and Test% measured after full retraining; mean $\pm$ std.

γ	Depth	Widths	Params (M)	$H_{\mathcal{V}} \downarrow$	Test% $\uparrow$
0.05	5	$[128, 256, 512, 512, 512]$	15.2	0.64±0.02	78.50±0.30
0.10	4	$[64, 128, 256, 512]$	9.5	0.68±0.02	79.20±0.22
0.20	4	$[64, 128, 256, 256]$	7.5	0.72±0.02	**79.50±0.20**
0.40	3	$[64, 128, 128]$	3.5	1.05±0.03	76.80±0.35
0.80	2	$[64, 64]$	1.8	1.72±0.05	71.50±0.55