SENSOR PRE-ACQUISITION NEURAL ARCHITECTURE SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimal neural architectures depend on input data acquisition, yet neural architecture search (NAS) typically fixes acquisition, excluding alternative acquisition–architecture pairs. Changing pre-acquisition parameters changes observations, preventing direct backpropagation through acquisition choices. Black-box joint search avoids this requirement but incurs substantial cost from training candidate pairs separately. We propose SPANAS, a NAS framework exploiting ordinal acquisition parameters to jointly search configurations and architectures without a differentiable acquisition model. A Rao–Blackwellized loss contrast between adjacent measured levels yields an unbiased within-interval selector gradient under fixed shared weights and generator parameters, without interpolating observations. A structured configuration representation conditions a hypernetwork over a weight-sharing supernetwork. Refinement and retraining verification produce an empirical accuracy–storage front. We evaluate SPANAS on FMCW-radar marshalling-signal recognition and a public wearable inertial benchmark. In both domains, SPANAS achieves higher mean hypervolume and mean best recognition performance than the evaluated black-box campaigns, raising hypervolume by 11% on radar and 7% on IMU over the strongest campaign. Its front also dominates every evaluated differentiable NAS result in both domains. On IMU, SPANAS achieves a higher macro-F1 than the strongest domain-specific architecture at the placement recommended by prior work with 5.7× less storage.

1 INTRODUCTION

Neural architecture search (NAS) increasingly designs the networks behind intelligent sensing systems (Lin et al., 2020; Bian et al., 2025; Zales et al., 2026), and as those systems grow more capable the demand extends to acquiring sensor data in the form the network can exploit best (Ballard et al., 2021; Njor et al., 2025; Khan & Gauch, 2025). NAS typically treats *pre-acquisition parameters*—settings fixed before recording, such as radar samples per chirp or wearable sensor placement—as given. Yet the best configuration depends on the architecture that processes its data (Enokibori et al., 2025), and vice versa. We also observe this dependence in our own experiments (Fig. 1, Sec. 4.2, and Appendix A). Changing acquisition selects different observations, preventing direct backpropagation through that choice. Black-box joint search can optimize both variables, but trains candidate pairs repeatedly.

In this work, we propose SPANAS (Sensor Pre-Acquisition Neural Architecture Search), a NAS framework that optimizes pre-acquisition parameters jointly with the network architecture in a single search. SPANAS exploits the ordinal structure of acquisition levels, assigning each parameter a learnable coordinate that selects between adjacent measured levels. Following the Rao–Blackwellization (RB) of single-sample score estimators (Titsias & Lázaro-Gredilla, 2015; Liu et al., 2019b; Paulus et al., 2021), the difference between the losses measured at those two levels gives an unbiased, variance-reduced gradient for the coordinate. In addition, we encode the selected pre-acquisition configuration in a structured representation whose per-axis and interaction components condition a hypernetwork that generates the architecture over a weight-sharing supernetwork, so that the network architecture is optimized to follow changes in the pre-acquisition parameters. Furthermore, the staged pipeline of search, refinement, and verification not only improves the searched pair but also returns an accuracy–storage front rather than a single network, so that SPANAS operates as a multi-objective optimization method.

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.

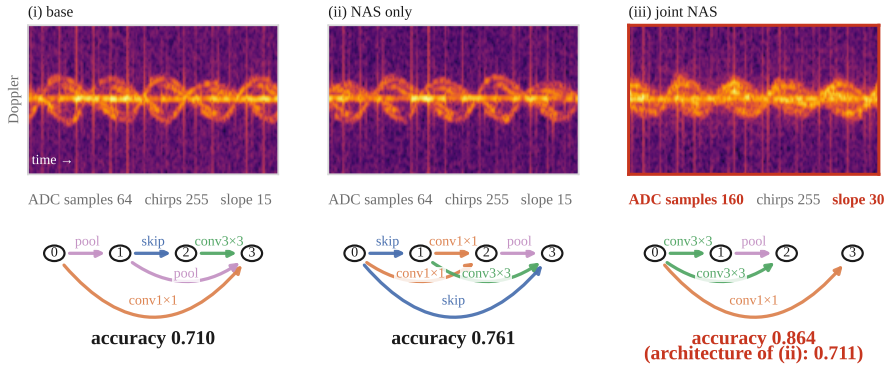


Figure 1: Configuration–architecture dependence illustrated by three pairs from the configuration-wise radar experiments (Appendix A): (i) a base pair, (ii) the architecture optimized at the same configuration (NAS only), and (iii) the jointly optimized configuration–architecture pair (joint NAS; changed parameters in red). Top panels show one gesture; the scores summarize recognition accuracy for each pair.

We evaluate on FMCW-radar marshalling-signal recognition and a public wearable inertial benchmark. Code and configurations will be released upon publication.

Contributions.

- We propose joint acquisition–architecture search without a differentiable acquisition model, adapting Rao–Blackwellized score estimation to ordinal acquisition parameters to obtain an adjacent-level estimator whose selector gradient is unbiased within an active interval under fixed shared weights and generator parameters.
- SPANAS conditions a hypernetwork on an identifiable axis-wise representation, sharing architecture information across configurations to optimize pre-acquisition parameters and the architecture jointly.
- We evaluate on two sensing modalities, a newly collected radar dataset and a public IMU benchmark, outperform black-box, categorical, and domain-specific baselines on verified accuracy–storage fronts, and analyze the role of each component through ablations.

2 Background and Related Work

One-shot differentiable NAS. One-shot NAS trains a single weight-sharing supernetwork in which every search edge e holds a set of candidate operations $\mathcal{O}_e$; an architecture is a choice of one operation per edge, and all architectures share the weights W (Pham et al., 2018; Bender et al., 2018). Differentiable NAS, introduced by DARTS, attaches logits $\boldsymbol{\alpha}_e$ to each edge, relaxes the choice to the mixture $\sum_{o\in\mathcal{O}_e} \mathrm{softmax}(\boldsymbol{\alpha}_e)_o\, o(x)$, and updates logits and weights by alternating gradient steps on the bilevel problem

$$\min_{\boldsymbol{\alpha}} \ \ell_{\mathrm{val}}\big(W^\star(\boldsymbol{\alpha}), \boldsymbol{\alpha}\big) \quad \text{s.t.} \quad W^\star(\boldsymbol{\alpha}) \in \arg\min_{W} \ \ell_{\mathrm{tr}}(W, \boldsymbol{\alpha}) \tag{1}$$

(Liu et al., 2019a). We use the sampling form of this relaxation: one hard operation per edge is executed in the forward pass and the gradient is propagated through a Gumbel-softmax surrogate (Dong & Yang, 2019; Xie et al., 2019). MODNAS generates the architecture logits from a preference vector through a hypernetwork so that one search covers a whole accuracy–cost front (Sukthanker et al., 2025); we also generate the logits through a hypernetwork but condition it on the acquisition configuration. Once-for-All separates supernetwork training from architecture specialization for deployment constraints (Cai et al., 2020); we likewise separate the search from refinement and verification stages, which produce the front.

Differentiable search beyond operations. Gradient methods already search ordered discrete choices such as kernel size, channel width, input resolution, bit-width, and feature-extraction settings, but each relaxes the levels inside one recorded tensor (Stamoulis et al., 2019; Wan et al., 2020;

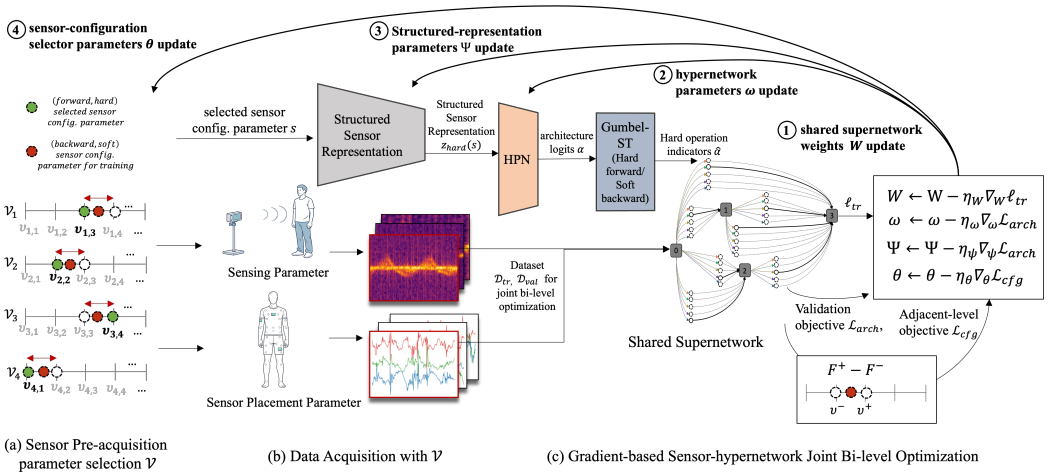


Figure 2: Overview of SPANAS: ordinal selection between adjacent measured levels (a), acquisition of the selected dataset (b), and joint bi-level optimization (c), in which a structured representation conditions a hypernetwork over a shared supernetwork and the validation and adjacent-level objectives update the parameters.

Uhlich et al., 2020; Shi et al., 2025; Njor et al., 2025) or treats them as unordered categories (Jang et al., 2017); a pre-acquisition choice decides which signal is acquired, so the data sets at two configurations are separate measurements that no relaxation of the input connects.

Gradient estimation for discrete choices. Score-function estimators optimize an expected loss without differentiating through the discrete choice, but can have high variance (Williams, 1992; Mohamed et al., 2020). Rao–Blackwellization replaces a sampled score by its conditional expectation and reduces variance without introducing bias (Titsias & Lázaro-Gredilla, 2015; Liu et al., 2019b; Paulus et al., 2021). For independent Bernoulli choices, integrating out one bit yields a contrast between its two endpoint losses. We apply this contrast to adjacent measured acquisition levels (Sec. 3.2); Appendix D.1 gives the score-function identity and derivation.

Black-box joint search. Without a gradient, acquisition and architecture variables can be searched jointly by multi-objective Bayesian optimization or evolutionary search (Knowles, 2006; Daulton et al., 2021; Deb et al., 2002; Bergstra & Bengio, 2012) at one full training per evaluation; these are our black-box competitors in Sec. 4.

Sensor design and configuration. Sensor design is increasingly framed as co-design with the computation that processes the measurements (Ballard et al., 2021): differentiable optics and pixel-layout methods back-propagate through a differentiable acquisition model, a simulator of the acquisition (Wetzstein et al., 2020; Tseng et al., 2021; Sommerhoff et al., 2024), and pre-acquisition parameters have been tuned for a fixed recognizer by Bayesian optimization (Kim & Kim, 2024). Pre-acquisition parameters are known to affect recognition performance (Tsukamoto et al., 2023; Park et al., 2026; Ayaz et al., 2025), and the preferred parameters can depend on the architecture that will process the data (Enokibori et al., 2025), yet existing sensing-oriented architectures and configuration-selection methods (Mutegeki et al., 2021; Lu et al., 2022; Ordóñez & Roggen, 2016; Malekzadeh et al., 2021; Sah & Ghasemzadeh, 2024) optimize only one of the two. SPANAS extends this search to acquisition choices without a differentiable acquisition model.

3 Method

3.1 Problem formulation

A pre-acquisition configuration assigns one level to every pre-acquisition parameter axis, such as a sensing parameter or a sensor placement coordinate (Fig. 2a). Each searched axis $q \in \{1, \dots, Q\}$ has $K_q \geq 2$ admissible levels $\nu_{q,0} < \cdots < \nu_{q,K_q-1}$. Pre-acquisition parameters of a sensing device are predominantly ordinal (Appendix B); SPANAS targets this setting and uses only

the order of the levels, so a configuration is represented by its rank vector $\mathbf{s} = (s_1, \ldots, s_Q)$ with $s_q \in \{0, \ldots, K_q - 1\}$; $\mathcal{S}$ denotes the set of all rank vectors. The method optimizes over measured observations: it requires training and validation observations $(\mathcal{D}_{\mathbf{s}}^{\mathrm{tr}}, \mathcal{D}_{\mathbf{s}}^{\mathrm{val}})$ at every setting it evaluates, and needs no acquisition model to predict or interpolate them. Observations may be recorded before the search, as in our experiments, or acquired whenever the search evaluates a setting.

We jointly select the acquisition configuration and architecture, making the training and validation data in Eq. 1 search-dependent:

$$\min_{\mathbf{s}\in\mathcal{S},\,\boldsymbol{\alpha}} \ \ell_{\mathrm{val}}\big(W^\star(\mathbf{s},\boldsymbol{\alpha}),\boldsymbol{\alpha};\mathcal{D}_{\mathbf{s}}^{\mathrm{val}}\big) + \lambda\, C(\boldsymbol{\alpha}) \quad \text{s.t.} \quad W^\star(\mathbf{s},\boldsymbol{\alpha}) \in \arg\min_{W} \ \ell_{\mathrm{tr}}(W,\boldsymbol{\alpha};\mathcal{D}_{\mathbf{s}}^{\mathrm{tr}}). \tag{2}$$

Here $\boldsymbol{\alpha}$ denotes architecture logits, W shared weights, $C(\boldsymbol{\alpha}) \in [0,1]$ normalized deployed storage, and $\lambda \geq 0$ a fixed regularization coefficient. Because $\mathcal{D}_{\mathbf{s}}$ selects separate measurements, Eq. 2 admits no direct pathwise gradient through acquisition. We construct a selector gradient, condition the architecture on its choice, and optimize both with shared weights.

3.2 Pre-acquisition selector and its gradient

Each pre-acquisition parameter axis q carries one learnable scalar θ_q from which its rank s_q is drawn; the draws over all axes define the configuration distribution $p_{\boldsymbol{\theta}}$. The scalar is mapped to a continuous position r_q on the rank grid of the axis:

$$\begin{aligned} m_q &= \mathrm{sigm}(\theta_q), \quad r_q = (K_q - 1)\, m_q, \\ j_q &= \min\{\lfloor r_q \rfloor, K_q - 2\}, \quad \delta_q = r_q - j_q, \end{aligned} \tag{3}$$

so that r_q ranges over $[0, K_q - 1]$; the integer part j_q names the *active cell* $[j_q, j_q{+}1]$ between two adjacent ranks (the minimum in $j_q = \min\{\lfloor r_q \rfloor, K_q - 2\}$ keeps the cell inside the grid when r_q reaches the top rank) and the local coordinate $\delta_q \in [0,1]$ specifies the position within the active interval, with $\delta_q = 1$ at the uppermost rank. A hard rank is drawn by stochastic rounding, $b_q \sim \mathrm{Bernoulli}(\delta_q)$, $s_q = j_q + b_q$: the lower endpoint $k_q^- := j_q$ with probability $1 - \delta_q$, the upper endpoint $k_q^+ := j_q{+}1$ with probability δ_q (Fig. 3). The joint law factorizes as $p_{\boldsymbol{\theta}}(\mathbf{s}) = \prod_q \delta_q^{b_q}(1-\delta_q)^{1-b_q}$ on the local support $\prod_q \{k_q^-, k_q^+\}$, so each axis moves probability mass only between its two adjacent levels, and every forward pass uses data acquired at one of them.

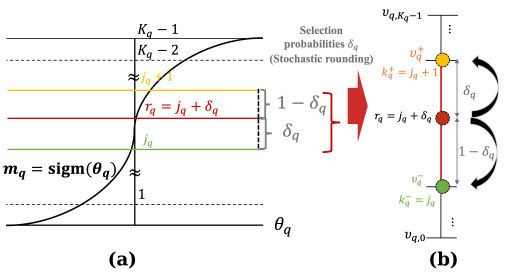


Figure 3: Ordinal selection on one axis. (a) θ_q is mapped to a continuous position $r_q = j_q + \delta_q$ on the level grid. (b) Stochastic rounding picks the lower or upper neighbouring measured level with probability $1 - \delta_q$ or δ_q; no level is ever interpolated.

The selector is updated from measured validation costs. At configuration $\mathbf{s}$, the sampled architecture $\widehat{\mathbf{a}}$ (Sec. 3.3) is scored on B validation mini-batches:

$$F(\mathbf{s}, \widehat{\mathbf{a}}) = \mathrm{sg}\Big[\tfrac{1}{B}\textstyle\sum_{r=1}^{B} \ell_{\mathrm{val}}^{(r)}\big(W, \widehat{\mathbf{a}}; \mathcal{D}_{\mathbf{s}}^{\mathrm{val}}\big) + \lambda\, C(\widehat{\mathbf{a}})\Big], \tag{4}$$

Here, $\mathrm{sg}[\cdot]$ denotes stop-gradient. During the selector update, the shared weights W and the representation and hypernetwork parameters Ψ and ω (Sec. 3.3) are held fixed; gradients flow only through the selection probabilities. Write $F(\mathbf{b}, \xi)$ for this cost at $\mathbf{s} = \mathbf{j} + \mathbf{b}$, where ξ collects mini-batch and architecture-sampling randomness, independently of the rounding bits. The endpoint costs $F_q^-(\mathbf{b}_{-q}, \xi)$ and $F_q^+(\mathbf{b}_{-q}, \xi)$ set b_q to 0 and 1, respectively, while keeping the other bits fixed. Conditional on $(\mathbf{b}_{-q}, \xi)$, averaging over b_q gives $(1-\delta_q)F_q^- + \delta_q F_q^+$, whose derivative with respect to δ_q is $F_q^+ - F_q^-$. Thus the loss contrast differentiates the selection probabilities without differentiating or interpolating observations. The expected objective is $J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{b},\xi}[F(\mathbf{b}, \xi)]$. We implement its selector update with the detached surrogate, the adjacent-level objective of Fig. 2(c),

$$\mathcal{L}_{\mathrm{cfg}} = \sum_q \mathrm{sg}\big[F_q^+(\mathbf{b}_{-q}, \xi) - F_q^-(\mathbf{b}_{-q}, \xi)\big]\ \delta_q(\theta_q), \tag{5}$$

so that a lower upper-endpoint cost raises the probability of rounding up. Differentiating $\mathcal{L}_{\text{cfg}}$ with respect to θ_q applies the chain rule through the selection probability. Inside a cell, j_q is constant and $\partial\delta_q/\partial\theta_q = (K_q - 1)m_q(1 - m_q)$ (Eq. 3), giving, with W, ω, Ψ fixed,

$$\frac{\partial J}{\partial \theta_q} = (K_q - 1)\, m_q(1 - m_q)\, \mathbb{E}_{\mathbf{b}_{-q},\xi}\big[F_q^+(\mathbf{b}_{-q}, \xi) - F_q^-(\mathbf{b}_{-q}, \xi)\big]. \tag{6}$$

The stochastic gradient of $\mathcal{L}_{\text{cfg}}$ is an unbiased estimator of this expected gradient within the active interval, with variance no larger than the score-function estimator (Proposition 1, Appendix D.1).

The two endpoint costs of every axis are measured by a base-and-flip construction: one base configuration and its single-axis neighbours are evaluated with shared architecture-sampling noise (common random numbers) and independent validation mini-batches to couple the endpoint evaluations; this coupling is distinct from the Rao–Blackwellization step. At each endpoint, we recompute the configuration representation and architecture logits and sample an architecture from the resulting distribution, while holding W, Ψ, ω fixed. Each contrast accounts for interactions with the sampled levels of the other axes within the active support. The selector learning rate controls the coordinate update size. After the search, the discrete pre-acquisition rank of axis q is $s_q^\star = \text{clip}(\lfloor r_q + \frac{1}{2} \rfloor, 0, K_q - 1)$.

3.3 Configuration-conditioned architecture

A free logit table learns each configuration independently. Instead, we share information through axis-wise effects while retaining configuration-specific interactions, following an analysis-of-variance structure (Fisher, 1935), in the structured representation of Fig. 2(c):

$$\mathbf{Z}_\mathbf{s} = \boldsymbol{\mu} + \sum_q \boldsymbol{\phi}^{\text{eff}}_{q,s_q} + \mathbf{R}^{\text{eff}}_\mathbf{s} \in \mathbb{R}^d, \tag{7}$$

Here $\boldsymbol{\mu}$ is a baseline shared by all configurations, $\boldsymbol{\phi}^{\text{eff}}_{q,s_q}$ the effect of level s_q of axis q, shared by every configuration containing it, and $\mathbf{R}^{\text{eff}}_\mathbf{s}$ the interaction left over for the specific configuration; all are computed from learnable tables Ψ. Each axis-level vector is reused by every configuration containing that level, so updates from different configurations train a shared component, while the interaction retains joint effects beyond these individual contributions. Centering makes each axis effect zero-mean, and projection removes constant and single-axis components from the interaction, so the three parts cannot absorb one another. Appendix C.1 gives the construction for each domain (one interaction over the three radar axes; one per sensor slot on IMU), and Appendix D.2 proves that the decomposition is unique under these constraints. A hypernetwork h_ω (HPN in Fig. 2) (Ha et al., 2017) maps $\mathbf{Z}_\mathbf{s}$ of Eq. 7 to the logits $\boldsymbol{\alpha}(\mathbf{s}) = h_\omega(\mathbf{Z}_\mathbf{s})$ of Eq. 1, one logit vector for each search edge. We evaluate the contribution of representation centering through ablation studies in Sec. 4.4.

Architectures $\widehat{\mathbf{a}}$ are sampled from $\boldsymbol{\alpha}(\mathbf{s})$ by straight-through Gumbel-softmax (Gumbel-ST) as in Sec. 2, and the generator is trained on the validation objective, the validation loss of the sampled architecture plus the expected storage cost under the sampling distribution, estimated with the soft assignment $\boldsymbol{\pi}$ that produced $\widehat{\mathbf{a}}$ (storage is additive over edges, so $\mathbb{E}_{\boldsymbol{\pi}}[C]$ is the $\boldsymbol{\pi}$-weighted sum of per-operation storage),

$$\mathcal{L}_{\text{arch}} = \ell_{\text{val}}\big(W, \widehat{\mathbf{a}}; \mathcal{D}^{\text{val}}_\mathbf{s}\big) + \lambda\, \mathbb{E}_{\boldsymbol{\pi}}[C], \tag{8}$$

the counterpart for (ω, Ψ) of the selector loss $\mathcal{L}_{\text{cfg}}$ of Eq. 5.

3.4 Joint acquisition–architecture objective and staged optimization

The relaxed joint objective. Combining the ordinal selector $p_{\boldsymbol{\theta}}$ over pre-acquisition configurations (Sec. 3.2) with the configuration-conditioned generator $\boldsymbol{\alpha}(\mathbf{s}) = h_\omega(\mathbf{Z}_\mathbf{s})$ (Sec. 3.3), we express the problem of Sec. 3.1 as the relaxed bilevel problem over the selector, hypernetwork, and representation parameters,

$$\min_{\boldsymbol{\theta},\omega,\Psi}\ \mathbb{E}_{\mathbf{s}\sim p_{\boldsymbol{\theta}}}\Big[\ell_{\text{val}}\big(W^\star, \boldsymbol{\alpha}(\mathbf{s}); \mathcal{D}^{\text{val}}_\mathbf{s}\big) + \lambda\, C(\boldsymbol{\alpha}(\mathbf{s}))\Big], \tag{9}$$

where $W^\star(\omega, \Psi)$ solves the inner problem

$$W^\star \in \arg\min_W\ \mathbb{E}_{\mathbf{s}_u\sim\text{Unif}(\mathcal{S})}\big[\ell_{\text{tr}}(W, \boldsymbol{\alpha}(\mathbf{s}_u); \mathcal{D}^{\text{tr}}_{\mathbf{s}_u})\big], \tag{10}$$

and $\ell(\cdot, \boldsymbol{\alpha})$ is the loss of the architecture sampled from $\boldsymbol{\alpha}$. The two losses already defined address this outer level (Fig. 2): $\mathcal{L}_{\text{arch}}$ of Eq. 8 moves (ω, Ψ) and $\mathcal{L}_{\text{cfg}}$ of Eq. 5 moves $\boldsymbol{\theta}$.

Uniform inner sampling prevents configurations favored early by the selector from monopolizing shared-weight updates. Algorithm 1 summarizes the stages; Algorithm 2 in Appendix C gives the full procedure.

Staged optimization. In Phase A each iteration takes three gradient steps (Algorithm 1, lines 3–6): W on Eq. 10 at a uniformly drawn configuration along a path from the conditioned generator; (ω, Ψ) on $\mathcal{L}_{\text{arch}}$ at $\mathbf{s} \sim p_{\boldsymbol{\theta}}$; and $\boldsymbol{\theta}$ on $\mathcal{L}_{\text{cfg}}$ from the base-and-flip endpoint costs. During warm-up $\boldsymbol{\theta}$ is fixed and the storage term is off.

Algorithm 1 SPANAS pipeline

1: **A: joint search** over $(W, \omega, \Psi, \boldsymbol{\theta})$
2: **repeat**
3: $\quad \mathbf{s}_u \sim \text{Unif}(\mathcal{S})$; $W \leftarrow$ step on $\ell_{\text{tr}}(\mathbf{s}_u)$
4: $\quad \mathbf{s} \sim \text{Unif}(\mathcal{S})$ in warm-up, else $\mathbf{s} \sim p_{\boldsymbol{\theta}}$
5: $\quad (\omega, \Psi) \leftarrow$ step on $\mathcal{L}_{\text{arch}}(\mathbf{s})$
6: $\quad$ After warm-up: $F_q^{\pm} \leftarrow$ base-and-flip; $\boldsymbol{\theta} \leftarrow$ step on $\mathcal{L}_{\text{cfg}}$
7: **until** budget exhausted
8: $\mathbf{s}^\star \leftarrow \text{discretize}(\boldsymbol{\theta})$
9: **B:** $(W', \boldsymbol{\alpha}) \leftarrow$ init; progressive search at $\mathbf{s}^\star$
10: $\mathcal{C} \leftarrow$ top-k frequent cells of the $\arg\max \boldsymbol{\alpha}$ trajectory
11: **C:** $\forall a \in \mathcal{C}$: retrain, evaluate
12: **return** front of $\mathcal{C}$

Once Phase A selects $\mathbf{s}^\star$, Phase B re-initializes the supernetwork weights and the architecture logits at this fixed configuration and searches again with plain logits, since the hypernetwork has nothing to condition on once the configuration is fixed (line 9). The Phase-A weights were trained evenly over all configurations, which served the acquisition search but leaves them unspecialized for choosing the architecture at one configuration. Refinement then decides the cell one edge at a time: after each training round it freezes the edge whose best operation is clearest, the one with the largest probability gap over its second-best, and keeps training the others until all edges are fixed; the shortlist $\mathcal{C}$ keeps up to k distinct cells by their frequency along the $\arg\max$ trajectory (line 10; Appendix C.2). Because rankings under shared weights are unreliable (Yu et al., 2020), Phase C takes each candidate cell out of the supernetwork and trains it from scratch as a standalone network under the common evaluation protocol (line 11); these verified scores form the accuracy–storage front.

4 Experiments

Sec. 4.1 describes the experimental setup; Sec. 4.2 compares SPANAS with black-box joint search on verified accuracy–storage fronts; Sec. 4.3 compares it with existing differentiable NAS methods, extended directly to the acquisition axes or applied at black-box-selected settings, and with domain-specific IMU architectures; Sec. 4.4 ablates the components.

4.1 Experimental setup

Domains and search spaces. Radar comprises seven aircraft-marshalling gestures, following the recording protocol of Müller et al. (2023), newly recorded from five participants with an FMCW system. ADC samples, chirps per frame, and chirp slope each have three measured levels, yielding 27 acquisition configurations, written as ADC samples/chirps/slope. IMU comprises 14 activities from 12 participants at 396 candidate sensor locations (Tsukamoto et al., 2023). We assign three selection slots to the upper-front panel and one to the lower-front panel, and search their row and column coordinates; a placement is written as its four location indices. The three radar axes share one interaction term; IMU uses a row–column interaction per slot. Both domains use a four-node, six-edge cell: five NAS-Bench-201 operations on radar and eight temporal operations on IMU, including recurrent and attention operators (Dong & Yang, 2020; Li et al., 2021). Radar represents sensing parameters that configure the sensor hardware before recording, the most common form of pre-acquisition choice; IMU tests whether the same search extends from such parameters to sensor placement, a physical choice over a two-dimensional grid. Within each domain, all compared joint-search methods use the same acquisition choices, cell operations, and recordings. Appendix E gives the full experimental details, including the search spaces, data protocols, and gesture and activity classes.

Comparisons. Black-box (BB) campaigns jointly search configurations and cells with $N{=}100$ candidate evaluations: qNParEGO (Knowles, 2006), qNEHVI (Daulton et al., 2021), NSGA-II (Deb

Table 1: Verified Pareto-front quality (mean±std over three search seeds). "Best" is the highest verified score per run. Campaign budget: $N=100$.

Radar	HV↑	GD↓	IGD↓	Best acc.↑
qNParEGO (Knowles, 2006)	0.308±0.027	0.096±0.034	0.079±0.030	0.889±0.025
qNEHVI (Daulton et al., 2021)	0.230±0.148	0.102±0.107	0.127±0.110	0.786±0.151
NSGA-II (Deb et al., 2002)	0.230±0.125	0.104±0.145	0.113±0.121	0.794±0.112
Random search (Bergstra & Bengio, 2012)	0.299±0.017	0.116±0.066	0.090±0.021	0.859±0.015
SPANAS (ours)	**0.342±0.015**	**0.011±0.016**	**0.023±0.031**	**0.908±0.013**
IMU	HV↑	GD↓	IGD↓	Best F1↑
qNParEGO (Knowles, 2006)	0.177±0.029	0.056±0.053	0.075±0.017	0.782±0.035
qNEHVI (Daulton et al., 2021)	0.224±0.038	0.037±0.048	0.063±0.035	0.828±0.047
NSGA-II (Deb et al., 2002)	0.163±0.030	0.046±0.076	0.085±0.019	0.762±0.026
Random search (Bergstra & Bengio, 2012)	0.203±0.053	0.048±0.032	0.058±0.019	0.807±0.054
SPANAS (ours)	**0.240±0.027**	**0.019±0.023**	**0.037±0.019**	**0.845±0.023**

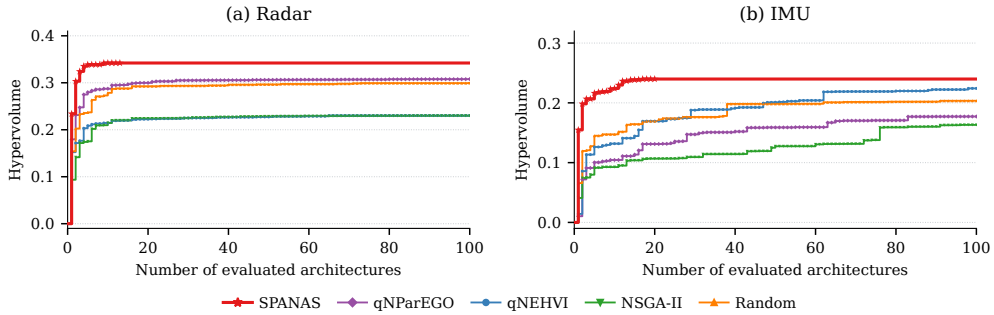


Figure 4: HV progression on (a) radar and (b) IMU, averaged over three searches. Campaign curves are validation-proxy histories calibrated to verified endpoints; SPANAS accumulates verified candidates (Phases A–C) in archive order.

et al., 2002), and random search (Bergstra & Bengio, 2012). We also compare four differentiable NAS methods (Sec. 4.3) and published IMU architectures at the recommended placement (Mutegeki et al., 2021; Lu et al., 2022; Ordóñez & Roggen, 2016).

Evaluation. Reported candidates are independently retrained under a common domain-specific protocol: 50 epochs across three participant-held-out retraining folds for radar accuracy; 150 epochs with nine training, one validation, and two test participants for IMU macro-F1. Storage counts deployed parameters, using fp32 on radar and the benchmark's mixed-precision convention on IMU (Tsukamoto et al., 2023). We report hypervolume (HV), generational distance (GD), and inverted GD (IGD); Table 1 gives means and sample standard deviations over three searches. HV is the normalized objective-space area dominated by a front, so higher is better; lower GD and IGD indicate that a front lies closer to, and covers more of, the empirical reference front, so a front that leads on all three approaches the reference front itself. Candidate selection precedes final scoring. Appendix E specifies the normalization and reference points used for hypervolume, the seeds used for reproducibility, and the retraining settings.

4.2 Comparison with black-box search

Verified front quality. SPANAS attains the highest mean HV and mean best recognition score, and the lowest mean GD and IGD, among the evaluated campaigns in both domains (Table 1).

Radar. Mean HV increases from 0.308 for qNParEGO, the strongest campaign in Table 1, to 0.342 for SPANAS, an 11.1% relative increase in the dominated area. Mean best accuracy increases by 1.9 percentage points, from 0.889 to 0.908, while GD and IGD decrease from 0.096 and 0.079 to 0.011 and 0.023. The reported searches select 160/255/30, which also yields the highest observed accuracy

in independent 100-evaluation architecture searches run separately at each of the 27 configurations (Appendix A).

IMU. Likewise, SPANAS obtains mean HV 0.240 versus 0.224 for qNEHVI, the strongest campaign, a 7.0% relative increase in the dominated area, and mean best macro-F1 0.845 versus 0.828.

Deployment trade-offs. The verified fronts provide models at several storage levels: at the selected radar configuration, the pooled SPANAS front ranges from 0.789 accuracy at 284 KB to 0.918 at 1,343 KB, and on IMU it includes macro-F1 0.822 at 96 KB and 0.863 at 232 KB. In both domains the SPANAS front dominates the largest area of the accuracy–storage space among the evaluated methods (Table 1); Appendix F pairs every operating point with its architecture and acquisition setting.

HV progression. Fig. 4 tracks HV against the number of evaluated architectures; the SPANAS curve adds verified candidates of Phases A–C in archive order. Campaigns improve gradually over 100 evaluations, whereas the SPANAS mean curve passes the final HV of the strongest campaign after three verified candidates on radar and eleven on IMU, and reaches its final value after 9 and 17 candidates, so the front converges with far fewer evaluated architectures than a campaign's 100. Individual runs and the curve construction are in Appendix F.1, measured search times in Appendix F.2.

4.3 Comparison with NAS baselines

Domain-specific models. Table 2 compares SPANAS with three architectures designed for wearable activity recognition, retrained under the common protocol at the placement recommended by Tsukamoto et al. (2023). The strongest, iSPL-Inception, reaches macro-F1 0.856 at 1,325 KB; the joint-search front reaches 0.863 at 232 KB. The architecture-only control, which keeps the recommended placement and runs only Phases B and C of the SPANAS pipeline, reaches 0.826 at 188 KB and is dominated by the joint-search front at every storage level. As both forms share the architecture search, the gain comes from optimizing the placement jointly with the architecture, not from the architecture search alone.

Table 2: IMU domain comparison. Domain models and the architecture-only control use the recommended placement; SPANAS (ours) pools three joint-search fronts.

Model	KB↓	Macro-F1↑
iSPL-Inception (Mutegeki et al., 2021)	**1325**	**0.856**
MCH-CNN-GRU (Lu et al., 2022)	293	0.822
DeepConvLSTM (Ordóñez & Roggen, 2016)	305	0.721
Recommended placement	44	0.727
(Tsukamoto et al., 2023)	135	0.771
+ SPANAS architecture search	188	0.826
	240	0.815
SPANAS (ours)	**44**	**0.757**
	96	**0.822**
	127	**0.823**
	179	**0.842**
	232	**0.863**

Comparison with NAS methods. We compare SPANAS with four existing differentiable NAS methods, Gumbel-ST, the DARTS mixture, ProxylessNAS, and DrNAS, applied in two ways (Table 3): extended directly to the acquisition axes inside our pipeline, and applied to architecture search at the acquisition settings that black-box search ranks highest (BB). In the extended form, each method's choice mechanism replaces the ordinal selector, treating the levels of each axis as unordered categories; in the BB form, the acquisition setting is fixed first and only the architecture is searched. In both domains, every NAS result in Table 3 is dominated by a SPANAS point with higher accuracy at equal or lower storage.

Table 3: Comparison with differentiable NAS: highest verified accuracy (radar) or macro-F1 (IMU) and the storage of that model; SPANAS rows list four verified operating points, best first. Row construction: Appendix G.

Method	Radar Accuracy	Radar KB	IMU Macro-F1	IMU KB
NAS extended to the acquisition axes				
Gumbel-ST (Jang et al., 2017)	0.639	2293	0.847	429
DARTS mixture (Liu et al., 2019a)	0.632	1343	0.676	398
ProxylessNAS (Cai et al., 2019)	0.632	1343	0.819	315
DrNAS (Chen et al., 2021)	0.862	1343	0.766	315
Black-box-selected settings (BB) + NAS				
BB + Gumbel-ST	0.599	284	0.783	214
BB + DARTS	0.878	1453	0.777	616
BB + ProxylessNAS	0.621	394	0.828	345
BB + DrNAS	0.901	3351	0.754	344
SPANAS (ours)	**0.918**	**1343**	**0.863**	**232**
	0.913	**1234**	**0.842**	**179**
	0.867	**394**	**0.822**	**96**
	0.789	**284**	**0.757**	**44**

On IMU, the model found by SPANAS is the smallest and most accurate among those of the extended methods (232 KB, macro-F1 0.863; Gumbel-ST is next at 0.847 and 429 KB), and with

Table 4: Component ablations, each rerunning the pipeline with only the named component changed: HV, GD, and IGD.

	Radar			IMU		
Variant	HV↑	GD↓	IGD↓	HV↑	GD↓	IGD↓
Full (SPANAS)	0.325	0.030	0.108	0.263	0.020	0.065
w/o ordinal structure	0.257	0.099	0.122	0.245	0.065	0.053
w/o RB	0.325	0.030	0.108	0.190	0.101	0.085
w/o structured rep. (free lookup)	0.316	0.020	0.056	(not applicable)		
w/o conditioning (single architecture)	0.325	0.030	0.108	0.162	0.064	0.073
w/o centering and projection	0.325	0.030	0.108	0.229	0.018	0.079
w/o Phase C	0.262	0.048	0.220	0.242	0.025	0.073
w/o Phases B and C	0.236	0.076	0.174	0.145	0.076	0.094

black-box pre-selection DrNAS at 0.901 on radar and ProxylessNAS at 0.828 on IMU remain below SPANAS at smaller storage.

4.4 Ablations

Ordinal selector and Rao–Blackwellization. Replacing the ordinal selector by the categorical Gumbel-ST selector lowers HV in both domains (Table 4). The categorical relaxation discards the ordering of the levels, so the selector can no longer move between neighboring settings through measured loss differences; on radar it converges to 160/160/30 instead of 160/255/30. Removing Rao–Blackwellization leaves radar unchanged but lowers IMU HV markedly: the single-sample score gradient carries a score weight that diverges near the interval ends, so its variance is high, and on IMU, whose placement space is far larger, the selector settles at a worse placement.

Structured representation. The representation acts only during selection, since Phase B re-initializes the supernetwork with a free logit table. On radar, all three variants (free lookup, single architecture, and no centering and projection) still select 160/255/30 and therefore reproduce the Full candidates; over 27 configurations Phase A converges to the same selection with or without the representation, and the free lookup table differs only in its refinement. On IMU, where 396^4 placements rule out a free table, the representation determines the placement the search settles at. Removing conditioning, so that one architecture distribution serves every placement, lowers HV sharply and the best macro-F1 from 0.871 to 0.765: the loss contrasts that drive the selector no longer reflect placement-specific architectures, and the search collapses to a worse placement. Removing centering and projection also lowers HV: without these constraints the axis-wise and interaction components absorb one another, so the per-axis vectors no longer carry information shared across placements. The structured representation of Sec. 3.3 is therefore what makes the joint search work in our IMU experiment, whose placement space is large, and its identifiable structure keeps the shared information usable.

Pipeline stages. Deploying the refined cell without Phase C keeps the highest score but lowers HV in both domains, because the front shrinks to that single point and only verification turns the shortlist into a front; stopping after Phase A (w/o Phases B and C) lowers HV further, since the cell is never re-optimized at the selected setting. Overall, the components that shape the selector's gradient, the ordinal contrast, Rao–Blackwellization, and the structured representation, decide the outcome on IMU, whereas on radar Phase A reaches the same configuration under every variant except the categorical selector, and refinement and verification carry the front.

5 Conclusion

SPANAS makes pre-acquisition choices part of neural architecture search without requiring a differentiable acquisition model. Adjacent-level loss contrasts update the acquisition selector, and a structured representation conditions architecture generation on the selected measurements. Refinement and independent retraining turn the search output into a verified accuracy–storage front. Across radar and inertial sensing, SPANAS improves mean hypervolume and mean best recognition performance over the evaluated black-box campaigns, with lower front-distance metrics. The results support treating acquisition and architecture as coupled design choices, enabling compact sensing models to be selected from empirical trade-offs in the observations available to the system.

AI USE STATEMENT

AI assistants were used to aid and polish the writing, for literature retrieval and discovery, and to assist in proving mathematical claims. Generative AI tools were not used for research ideation or execution, for drafting sections of the paper, or for generating synthetic datasets. Mathematical derivations were re-derived by the authors. The authors take full responsibility for the research, including the verification of mathematical claims, experimental results, citations, and the final manuscript.

ETHICS STATEMENT

This study uses a public wearable inertial benchmark and radar recordings collected from five volunteers who provided informed consent. The radar data record marshalling gestures and are stored without participant names. Data collection and processing are described in Appendix E. The proposed method concerns sensor–architecture design for recognition systems. Applications involving human sensing should preserve participant privacy and follow the relevant consent and data-use requirements.

REPRODUCIBILITY STATEMENT

Appendix C details the algorithm, and Appendix D provides the theoretical assumptions and proofs. Dataset preparation, splits, search spaces, and evaluation protocols are specified in Appendix E; the verified operating points are listed in Appendix F. Execution settings are described in Appendix H. Code and experiment configurations will be released upon publication.

REFERENCES

Louis Atallah, Benny Lo, Rachel King, and Guang-Zhong Yang. Sensor positioning for activity recognition using wearable accelerometers. *IEEE Transactions on Biomedical Circuits and Systems*, 5(4):320–329, 2011.

Fahad Ayaz, Basim Alhumaily, Sajjad Hussain, Muhammad Ali Imran, Kamran Arshad, Khaled Assaleh, and Ahmed Zoha. Radar signal processing and its impact on deep learning-driven human activity recognition. *Sensors*, 25(3):724, 2025.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21524–21538, 2020.

Zachary Ballard, Calvin Brown, Asad M. Madni, and Aydogan Ozcan. Machine learning and computation-enabled intelligent sensor design. *Nature Machine Intelligence*, 3(7):556–565, 2021.

Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc V. Le. Understanding and simplifying one-shot architecture search. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 550–559, 2018.

Szabolcs Béres and László Hejjel. The minimal sampling frequency of the photoplethysmogram for accurate pulse rate variability parameters in healthy volunteers. *Biomedical Signal Processing and Control*, 68:102589, 2021.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

Sizhen Bian, Mengxi Liu, Vitor Fortes Rey, Daniel Geissler, and Paul Lukowicz. TinierHAR: Towards ultra-lightweight deep learning models for efficient human activity recognition on edge devices. *arXiv preprint arXiv:2507.07949*, 2025.

Julian Blank and Kalyanmoy Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020. doi: 10.1109/ACCESS.2020.2990567.

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

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HylxE1HKwS.

Ayan Chakrabarti. Learning sensor multiplexing design through back-propagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

Xiangning Chen, Ruochen Wang, Minhao Cheng, Xiaocheng Tang, and Cho-Jui Hsieh. DrNAS: Dirichlet neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2021.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2187–2200, 2021.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four GPU hours. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1761–1770, June 2019. doi: 10.1109/CVPR.2019.00186.

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

Yu Enokibori, Takahiro Sato, and Kenji Mase. Where are the best positions of IMUs for HAR? —investigation with four DNN models of different characteristics—. In *Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1089–1095, 2025. doi: 10.1145/3714394.3756334.

Ronald A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *ICLR*, 2018.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017.

Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding Zhao. Investigating the impact of multi-LiDAR placement on object detection for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2540–2549, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Aftab Khan, Nils Hammerla, Sebastian Mellor, and Thomas Plötz. Optimising sampling rates for accelerometer-based human activity recognition. *Pattern Recognition Letters*, 73:33–40, 2016.

Reeshad Khan and John Gauch. Learning to sense for driving: Joint optics-sensor-model co-design for semantic segmentation. *arXiv preprint arXiv:2512.20815*, 2025.

Sejung Kim and Jaeho Kim. Parameter optimization framework for enhancing radar-based material recognition. *IEEE Sensors Journal*, 24(24):42219–42229, 2024.

Joshua Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66, 2006. doi: 10.1109/TEVC.2005.851274.

Kai Kunze and Paul Lukowicz. Sensor placement variations in wearable activity recognition. *IEEE Pervasive Computing*, 13(4):32–41, 2014.

Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, and Yingyan Lin. HW-NAS-Bench: Hardware-aware neural architecture search benchmark. In *International Conference on Learning Representations*, 2021.

Ye Li, Lingdong Kong, Hanjiang Hu, Xiaohao Xu, and Xiaonan Huang. Is your LiDAR placement optimized for 3D scene understanding? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Ji Lin, Wei-Ming Chen, Yujun Lin, John Cohn, Chuang Gan, and Song Han. MCUNet: Tiny deep learning on IoT devices. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11711–11722, 2020.

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

Liyuan Liu, Chengyu Dong, Xiaodong Liu, Bin Yu, and Jianfeng Gao. Bridging discrete and backpropagation: Straight-through and beyond. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Runjing Liu, Jeffrey Regier, Nilesh Tripuraneni, Michael I. Jordan, and Jon McAuliffe. Rao-blackwellized stochastic gradients for discrete distributions. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4023–4031. PMLR, 2019b.

Pablo Ribalta Lorenzo, Lukasz Tulczyjew, Michal Marcinkiewicz, and Jakub Nalepa. Hyperspectral band selection using attention-based convolutional neural networks. *IEEE Access*, 8:42384–42403, 2020.

Limeng Lu, Chuanlin Zhang, Kai Cao, Tao Deng, and Qianqian Yang. A multichannel CNN-GRU model for human activity recognition. *IEEE Access*, 10:66797–66810, 2022. doi: 10.1109/ACCESS.2022.3185112.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. DANA: Dimension-adaptive neural architecture for multivariate sensor data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3), 2021. doi: 10.1145/3478074.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Ali Mosleh, Avinash Sharma, Emmanuel Onzon, Fahim Mannan, Nicolas Robidoux, and Felix Heide. Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7526–7535, 2020.

Leon Müller, Manolis Sifalakis, Sherif Eissa, Amirreza Yousefzadeh, Paul Detterer, Sander Stuijk, and Federico Corradi. Aircraft marshaling signals dataset of FMCW radar and event-based camera for sensor fusion. In *2023 IEEE Radar Conference (RadarConf23)*, pp. 1–6, 2023. doi: 10.1109/RadarConf2351548.2023.10149465.

Ronald Mutegeki, Alwin Poulose, and Dong-Seog Han. iSPLInception: An inception-ResNet deep learning architecture for human activity recognition. *IEEE Access*, 9:68985–69001, 2021. doi: 10.1109/ACCESS.2021.3078184.

Emil Njor, Colby Banbury, and Xenofon Fafoutis. Fast data aware neural architecture search via supernet accelerated evaluation. *arXiv preprint arXiv:2502.12690*, 2025.

Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016. doi: 10.3390/s16010115.

Do-Hyun Park, Min-Wook Jeon, and Hyoung-Nam Kim. Activity-dependent resolution adjustment for radar-based human activity recognition. *Signal Processing*, 243, 2026. doi: 10.1016/j.sigpro.2025.110456.

Max B. Paulus, Chris J. Maddison, and Andreas Krause. Rao–blackwellizing the straight-through Gumbel-Softmax gradient estimator. In *International Conference on Learning Representations*, 2021.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4095–4104, 2018.

Ramesh Kumar Sah and Hassan Ghasemzadeh. Minimum-cost sensor channel selection for wearable computing, 2024.

Yujia Shi, Emil Njor, Pablo Martínez-Nuevo, Sven Ewan Shepstone, and Xenofon Fafoutis. Data aware differentiable neural architecture search for tiny keyword spotting applications. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2025. doi: 10.1109/MLSP62443.2025.11204322.

Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics*, 37(4):114:1–114:13, 2018.

Hendrik Sommerhoff, Shashank Agnihotri, Mohamed Saleh, Michael Moeller, Margret Keuper, Bhaskar Choubey, and Andreas Kolb. Task driven sensor layouts—joint optimization of pixel layout and network parameters. In *IEEE International Conference on Computational Photography (ICCP)*, 2024. doi: 10.1109/ICCP61108.2024.10644474.

Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path NAS: Designing hardware-efficient ConvNets in less than 4 hours. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pp. 481–497, 2019.

Thomas Strypsteen and Alexander Bertrand. End-to-end learnable EEG channel selection for deep neural networks with Gumbel-softmax. *Journal of Neural Engineering*, 18(4):0460a9, 2021.

Rhea Sanjay Sukthanker, Arber Zela, Benedikt Staffler, Samuel Dooley, Josif Grabocka, and Frank Hutter. Multi-objective differentiable neural architecture search. In *International Conference on Learning Representations*, 2025.

Michalis K. Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. Differentiable compound optics and processing pipeline optimization for end-to-end camera design. *ACM Transactions on Graphics*, 40(2):18:1–18:19, 2021. doi: 10.1145/3446791.

Akihisa Tsukamoto, Naoto Yoshida, Tomoko Yonezawa, Kenji Mase, and Yu Enokibori. Where are the best positions of IMU sensors for HAR? —approach by a garment device with fine-grained grid IMUs—. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2023 ACM International Symposium on Wearable Computing*, pp. 445–450, 2023. doi: 10.1145/3594739.3610736.

George Tucker, Andriy Mnih, Chris J. Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NeurIPS*, 2017.

Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision DNNs: All you need is a good parametrization. In *International Conference on Learning Representations (ICLR)*, 2020.

Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, Peter Vajda, and Joseph E. Gonzalez. FBNetV2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12962–12971, 2020.

Gordon Wetzstein, Aydogan Ozcan, Sylvain Gigan, Shanhui Fan, Dirk Englund, Marin Soljačić, Cornelia Denz, David A. B. Miller, and Demetri Psaltis. Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836):39–47, 2020.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: Stochastic neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2019.

Chengyang Yan and Donald G. Dansereau. TaCOS: Task-specific camera optimization with simulation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2052–2062, 2025.

Kaicheng Yu, Christian Sciuto, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. In *International Conference on Learning Representations*, 2020.

Joseph Q. Zales, Pragya Sharma, and Mani Srivastava. CREST: Deployment-realistic hardware-in-the-loop NAS for embedded sensing systems. *arXiv preprint arXiv:2606.15004*, 2026.

Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi. LEAF: A learnable frontend for audio classification. In *International Conference on Learning Representations (ICLR)*, 2021.

Mingxing Zhu, Haoshi Zhang, Xiaochen Wang, Xin Wang, Zijian Yang, Cheng Wang, Oluwarotimi Williams Samuel, Shixiong Chen, and Guanglin Li. Towards optimizing electrode configurations for silent speech recognition based on high-density surface electromyography. *Journal of Neural Engineering*, 18(1):016005, 2021.

A RADAR ACQUISITION LANDSCAPE

Table 5 lists, for each of the 27 radar acquisition configurations, the best accuracy found by an independent 100-evaluation qNEHVI architecture search (Daulton et al., 2021) at that fixed configuration and the mean accuracy of the 100 architectures it evaluated; both are search-time evaluations. The highest best accuracy, 0.864, and the highest mean accuracy, 0.826 against at most 0.737 elsewhere, are both at 160 ADC samples, 255 chirps, 30 MHz/μs, the configuration SPANAS selects in all three searches (Table 10). Table 6 reads the same evaluations by architecture: for ten architectures evaluated at every configuration whose best configuration is not 160/255/30, it lists that configuration, the accuracy there, the accuracy at 160/255/30, and the accuracy lost by fixing the acquisition to 160/255/30. Five distinct configurations appear, and for two architectures the loss exceeds 0.19 (Fig. 1 draws on these rows). Read together, the two tables show the dependence that motivates joint search: the configuration that SPANAS selects is also the best configuration in these configuration-wise searches. Individual architectures, however, peak at different configurations, so neither the configuration nor the architecture can be fixed in advance without losing accuracy.

Table 5: Configuration-wise reference: for each of the 27 radar acquisition configurations, the best accuracy found by an independent 100-evaluation qNEHVI architecture search at that fixed configuration (search-time evaluation) and the mean accuracy of its 100 evaluated architectures. Rows are sorted by best accuracy and continue in the right half; bold: the configuration SPANAS selects.

Configuration (ADC/chirps/slope)	Best acc.	Mean acc.	Configuration (ADC/chirps/slope)	Best acc.	Mean acc.
160/255/30	**0.864**	**0.826**	64/160/15	0.698	0.554
64/255/30	0.810	0.704	160/255/15	0.644	0.575
160/64/30	0.783	0.737	160/160/15	0.634	0.614
64/255/15	0.761	0.729	256/255/30	0.605	0.528
64/255/45	0.752	0.658	160/64/15	0.599	0.526
256/160/15	0.751	0.623	160/160/45	0.598	0.520
256/64/45	0.726	0.647	64/64/45	0.586	0.463
64/160/30	0.722	0.676	256/255/45	0.564	0.429
64/160/45	0.718	0.682	64/64/30	0.553	0.474
160/64/45	0.713	0.655	256/160/30	0.544	0.467
256/255/15	0.710	0.614	256/64/15	0.506	0.417
256/160/45	0.703	0.526	64/64/15	0.459	0.420
160/160/30	0.702	0.622	256/64/30	0.429	0.384
160/255/45	0.702	0.632			

Table 6: Configuration-wise reference read by architecture: ten architectures evaluated at every configuration whose best configuration differs from 160/255/30 (search-time evaluation, as in Table 5), the accuracy at that configuration, the accuracy at 160/255/30, and the accuracy lost by fixing the acquisition to 160/255/30.

Arch.	Best config.	Acc.	At 160/255/30	Loss
8990	160/64/30	0.764	0.743	+0.021
4669	64/255/15	0.761	0.711	+0.050
12698	256/160/15	0.751	0.748	+0.003
1937	160/64/30	0.737	0.717	+0.021
4183	160/64/30	0.733	0.681	+0.052
10775	64/255/30	0.684	0.629	+0.055
7375	64/255/30	0.658	0.625	+0.033
3207	64/255/30	0.586	0.533	+0.053
1070	160/64/45	0.538	0.275	+0.263
8329	256/160/15	0.490	0.275	+0.215

B ORDINAL STRUCTURE OF PRE-ACQUISITION PARAMETERS

Table 7 lists acquisition parameters that published work varies or optimizes, across sensing modalities, with the structure of their admissible levels.

Fig. 5 shows how the observations change when one acquisition parameter moves by one level with the others fixed. On radar, moving one axis by one level changes the spectrogram gradually, and moving it by two levels changes it more; on the IMU, moving a sensor to a neighbouring site changes the waveform gradually, and the change grows with the grid distance. These observations motivate comparisons between neighboring acquisition levels. The change between adjacent levels is gradual but not differentiable: each level is a separate recording, so changing an acquisition parameter changes the observation without providing a path for backpropagation through the acquisition choice, and the amount of change per level differs across axes (on radar, a one-level move in chirps alters the spectrogram more than a one-level move in slope; Fig. 5). SPANAS therefore compares measured losses at adjacent levels rather than interpolating observations.

Of the 28 parameter families in Table 7, 19 are ordinal; the continuous ones are learned by gradient through a simulator or a differentiable front-end, or, for a black-box ISP, by zeroth-order search (Mosleh et al., 2020); SPANAS assumes neither. The categorical ones (waveform, named body sites, sensor model, electrode subsets, microphone type, colour filter and pixel layout) are choices among unordered alternatives, whether devices, named sites, or subsets, with no adjacent level to compare, and correspond to the categorical (Gumbel-ST) selection used as a baseline in Sec. 4.3 and in the w/o ordinal structure ablation of Sec. 4.4. Sensor placement appears in both forms: named body sites are categorical, positions on a garment, vehicle or electrode grid are ordinal in two dimensions.

C Method details

C.1 The structured representation as a parameter-table lookup

The representation $\mathbf{Z}_\mathbf{s}$ is assembled from a global vector, centered axis-level embeddings, and projected interaction entries indexed by the selected configuration (Eq. 7). These learnable tables belong to Ψ and do not depend on the input signal. Centering and projection remain in the computation graph, so gradients can reach both selected and unselected raw entries; the discrete configuration index itself does not carry a selector gradient. Radar uses one interaction group over its three acquisition axes, whereas IMU uses panel-specific row–column interactions within each sensor slot. For $d = 20$, the representations contain 740 and 65,540 parameters, respectively, excluding the hypernetwork and supernetwork.

Centering and projection. For the radar instance of Eq. 7, $\boldsymbol{\mu}$ is a baseline shared by all pre-acquisition configurations, $\boldsymbol{\phi}^{\text{eff}}_{q,k}$ the effect of level k of pre-acquisition parameter axis q, shared by every configuration that contains it, and $\mathbf{R}^{\text{eff}}_\mathbf{s}$ the coupling left over for $\mathbf{s}$ specifically. They are computed from raw parameters $\Psi = \{\boldsymbol{\mu}, \{\mathbf{E}^{\text{raw}}_q\}, \mathbf{R}^{\text{raw}}\}$ by centering each axis table across its levels and removing every one-axis marginal from the interaction tensor,

$$\boldsymbol{\phi}^{\text{eff}}_{q,k} = \mathbf{E}^{\text{raw}}_q[k] - \tfrac{1}{K_q}\textstyle\sum_\ell \mathbf{E}^{\text{raw}}_q[\ell], \qquad \mathbf{R}^{\text{eff}}_\mathbf{s} = \mathbf{R}^{\text{raw}}_\mathbf{s} - \textstyle\sum_q \overline{\mathbf{R}}^{\text{raw}}_{q,s_q} + (Q-1)\overline{\mathbf{R}}^{\text{raw}}, \tag{11}$$

with $\overline{\mathbf{R}}^{\text{raw}}_{q,k}$ the average of the raw interactions over pre-acquisition configurations with $s_q = k$ and $\overline{\mathbf{R}}^{\text{raw}}$ the average over all configurations, recomputed at every forward pass. The centering and projection make the three effective components identifiable for a fixed representation function (Proposition 2, Appendix D.2): an axis effect learned anywhere transfers to every configuration sharing the level, while couplings specific to one configuration remain expressible.

Slot-wise IMU interactions. For slot j assigned to panel c_j, let $\mathbf{T}_j = \mathbf{R}^{\text{raw}}_{j,c_j} \in \mathbb{R}^{U_j \times V_j \times d}$ be its row–column table. At the selected position (u_j, v_j) the representation uses the doubly centered vector

$$\mathbf{R}^{\text{eff}}_j(u_j, v_j) = \mathbf{T}_j[u_j, v_j] - \frac{1}{V_j}\sum_v \mathbf{T}_j[u_j, v] - \frac{1}{U_j}\sum_u \mathbf{T}_j[u, v_j] + \frac{1}{U_j V_j}\sum_{u,v} \mathbf{T}_j[u, v]. \tag{12}$$

Each mean is a d-vector. The representation sums these vectors over slots together with the centered row and column effects. With fixed panels, the centered panel effects are constant across searched configurations and can be absorbed into $\boldsymbol{\mu}_{\text{fixed}} = \boldsymbol{\mu} + \sum_j (\mathbf{E}_p[j, c_j] - \frac{1}{C}\sum_{c=1}^{C} \mathbf{E}_p[j, c])$, where C

Table 7: Pre-acquisition parameters varied or optimized in published work, by sensing front-end. A parameter is *ordinal* when its admissible levels are numerically ordered (the setting of this paper; 2-D marks positions on a grid), *categorical* when they are not, and *continuous* when it is optimized as a real-valued setting, through a simulator, a differentiable front-end, or zeroth-order search. Continuous settings that a device quantizes to a discrete list are counted as ordinal.

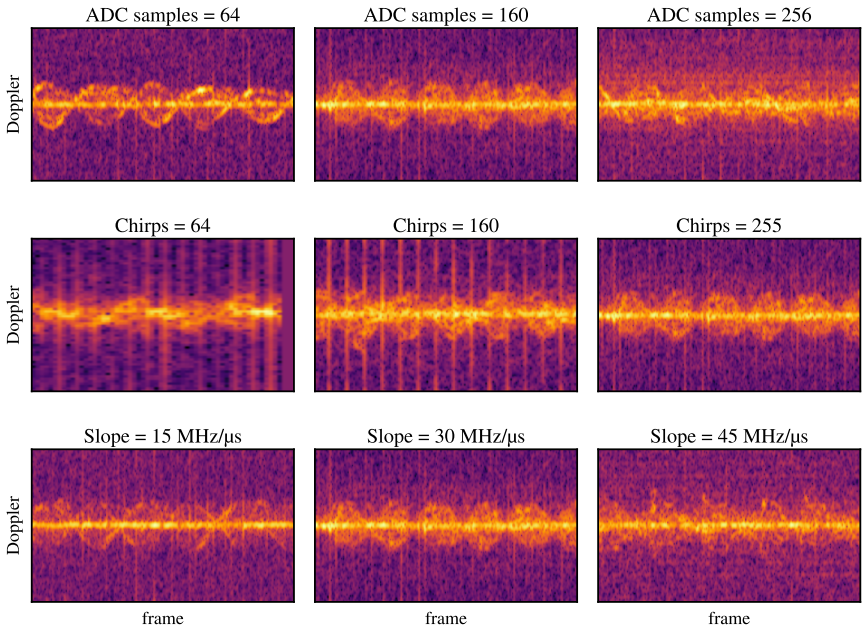
Parameter	Levels	Type	Reference
FMCW radar			
ADC samples per chirp	64–256	ordinal	this work
chirps per frame	64–255	ordinal	this work
frequency slope	15–45 MHz/μs	ordinal	this work
frame period, bandwidth, Tx/Rx count	lists	ordinal	Kim & Kim (2024)
waveform (FMCW / pulsed)	2	categorical	—
LiDAR			
mounting positions on the vehicle	roof grid	ordinal (2-D)	Hu et al. (2022) Li et al. (2024)
number of LiDARs	1–n	ordinal	Li et al. (2024)
Wearable IMU			
attachment site on a garment grid	396 sites	ordinal (2-D)	Enokibori et al. (2025)
displacement within a limb segment	positions	ordinal (spatial)	Kunze & Lukowicz (2014)
body site (ear, chest, wrist, ...)	7 sites	categorical	Atallah et al. (2011)
sampling rate	25–200 Hz	ordinal	Khan et al. (2016) Malekzadeh et al. (2021) Njor et al. (2025)
accelerometer / gyroscope range	$\pm$2–16 g, $\pm$250–2000 °/s	ordinal	—
number of sensors	1–n	ordinal	Enokibori et al. (2025)
sensor model	lists	categorical	—
Photoplethysmography			
sampling frequency	lists	ordinal	Béres & Hejjel (2021)
EEG and surface EMG			
EEG electrode subset	64 channels	categorical (subset)	Strypsteen & Bertrand (2021)
HD-sEMG electrode positions	120-channel grid	ordinal (2-D, subset)	Zhu et al. (2021)
Hyperspectral imaging			
spectral band subset	bands by wavelength	ordinal (subset)	Lorenzo et al. (2020)
Audio			
sampling rate, bit depth, gain	lists	ordinal	Njor et al. (2025)
window length, hop, mel bins	lists	ordinal	Shi et al. (2025)
filterbank centre frequencies, bandwidths	learnable	continuous	Zeghidour et al. (2021)
microphone count and spacing	lists	ordinal	—
microphone type	few	categorical	—
Camera			
resolution, frame rate, bit depth	lists	ordinal	Wetzstein et al. (2020)
exposure, gain, focal length / field of view	device lists or continuous	ordinal / continuous	Yan & Dansereau (2025)
ISP parameters (denoise, sharpen, tone)	mixed	continuous / discrete	Mosleh et al. (2020)
lens surfaces, phase-mask profile	learnable	continuous	Sitzmann et al. (2018) Tseng et al. (2021)
per-pixel colour filter, pixel layout	few	categorical	Chakrabarti (2016) Sommerhoff et al. (2024)

is the number of panels and $\mathbf{E}_p$ is the slot-wise panel-effect table (one d-vector per slot and panel). Thus Eq. 7 applies with $\mathbf{R}_{\mathbf{s}}^{\text{eff}}$ equal to the sum of these slot-wise vectors and baseline $\boldsymbol{\mu}_{\text{fixed}}$. No explicit cross-slot interaction table is stored; the hypernetwork nonlinearly combines the summed representation. This parameterization does not provide an arbitrary full interaction tensor over all slots. The row and column means of Eq. 12 are zero and applying this centering twice leaves it unchanged (Proposition 2).

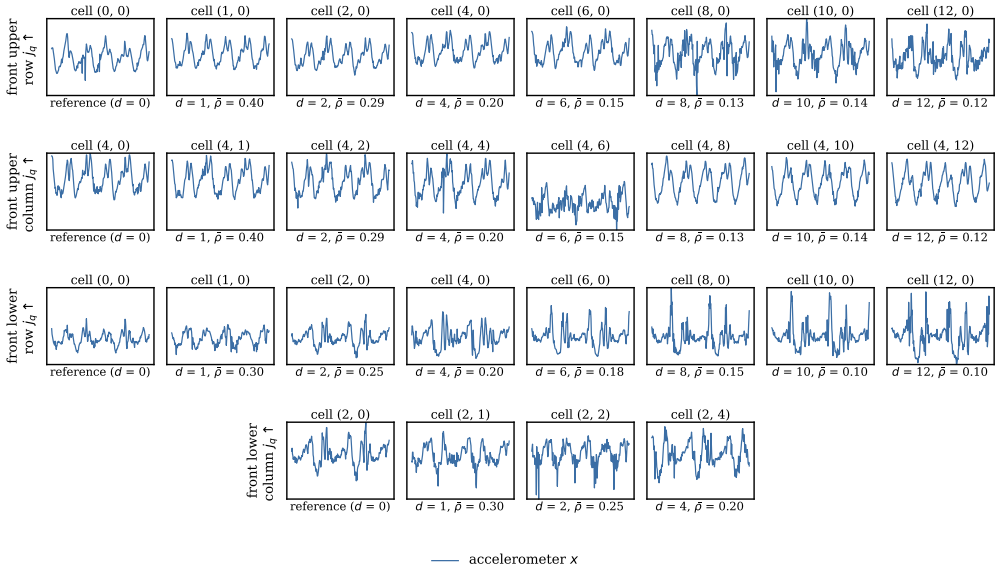
C.2 Staged search and verification (Phases A–C)

Algorithm 2 summarizes the complete joint-search, refinement, and verification pipeline. The discussion below details the refinement and independent verification stages.

Phase A trains the shared weights uniformly across acquisition configurations (Eq. 10). Phase B specializes architecture search at the selected configuration by re-initializing the supernetwork: the structured representation now produces one constant vector, so the hypernetwork is replaced by a free logit table $\boldsymbol{\alpha} \in \mathbb{R}^{|E|\times|\mathcal{O}|}$ while sampling, losses, and weight updates are unchanged. After a uniform warm-up, each round samples architectures with already committed edges fixed and commits the free edge with the largest top-1 margin; after $|E|$ rounds a short tail specializes the weights.



(a) Radar



(b) IMU

Figure 5: Observations at adjacent acquisition levels. (a) Radar: the micro-Doppler spectrogram of one gesture when one axis moves one level at a time with the other two fixed at 160 ADC samples, 255 chirps, 30 MHz/μs. (b) IMU: the accelerometer x channel of one walking window at grid cells of the front panels at increasing grid distance d from a reference cell, down a column and along a row; $\bar{\rho}$ is the panel-wide mean correlation between site pairs at distance d. A one-level move changes the observation gradually and the change grows with the distance.

We retain the k most frequent distinct, nonempty decoded cells along the recorded $\arg\max \boldsymbol{\alpha}$ trajectory, or all available cells if fewer than k occur (on radar, cells are reduced to a canonical form so that equivalent encodings count as one cell). Writing this set as $\mathcal{C}$, Phase C retrains each cell in $\mathcal{C}$ once under the common protocol. Candidate membership is fixed before this evaluation. The reported result is the empirical accuracy–storage front of these candidates. The reported front is

Algorithm 2 SPANAS pipeline (full procedure)

Require: Access to training and validation observations $(\mathcal{D}_{\mathbf{s}}^{\mathrm{tr}}, \mathcal{D}_{\mathbf{s}}^{\mathrm{val}})$ at evaluated settings $\mathbf{s}$, storage coefficient λ, search length N, warm-up N_{warm}, phase lengths N_U, N_R, N_T, candidate cap k

Phase A: joint search; initialize $W, \omega, \Psi, \boldsymbol{\theta}$; disable the storage term during warm-up
for $n = 1, \ldots, N$ **do**
  $\mathbf{s}_u \sim \mathrm{Unif}(\mathcal{S})$, $\mathbf{a} \sim \mathrm{GumbelST}(\boldsymbol{\alpha}(\mathbf{s}_u))$; update W with ℓ_{tr}
  if $n \le N_{\mathrm{warm}}$ **then**
    $\mathbf{s} \sim \mathrm{Unif}(\mathcal{S})$; update ω, Ψ with $\mathcal{L}_{\mathrm{arch}}$; keep $\boldsymbol{\theta}$ fixed
  else
    $\mathbf{s} \sim p_{\boldsymbol{\theta}}$; update ω, Ψ with $\mathcal{L}_{\mathrm{arch}}$
    Evaluate detached base-and-flip costs, recomputing the representation and architecture logits at each endpoint; update $\boldsymbol{\theta}$ with $\mathcal{L}_{\mathrm{cfg}}$
  end if
end for
$s_q^\star \leftarrow \mathrm{clip}(\lfloor (K_q - 1)\mathrm{sigm}(\theta_q) + \frac{1}{2} \rfloor, 0, K_q - 1)$ for each axis q
Phase B: refinement at $\mathbf{s}^\star$; initialize $W', \boldsymbol{\alpha}$; $\mathcal{F} \leftarrow \emptyset$
for $n = 1, \ldots, N_U$ **do**
  $\mathbf{a} \sim \mathrm{Unif}(\mathcal{A})$; update W'
end for
for round $r = 1, \ldots, |E|$ **do**
  for $n = 1, \ldots, N_R$ **do**
    Sample $\mathbf{a}$ with edges in $\mathcal{F}$ fixed and the rest by straight-through Gumbel-softmax
    Update W' on $\mathcal{D}_{\mathbf{s}^\star}^{\mathrm{tr}}$ and $\boldsymbol{\alpha}$ on held-out validation; record $\arg\max \boldsymbol{\alpha}$
  end for
  Commit the free edge with the largest top-1 margin to $\mathcal{F}$
end for
Train W' for N_T steps on the committed cell
$\mathcal{C} \leftarrow$ up to k most frequent distinct nonempty decoded cells in the trajectory (canonical form on radar)
Phase C: verification; retrain each candidate in $\mathcal{C}$ under the common protocol
return $\mathbf{s}^\star$, evaluated candidates $\mathcal{C}$, and their empirical accuracy–storage front

computed over all verified candidates, none of which is excluded; selecting a single deployment model is a separate validation-based step and is not done by taking the highest test score. The cap k bounds the number of candidates retrained in Phase C.

Weights inherited from a supernetwork are shared among all architectures, so the accuracy of a cell inside the supernetwork is a proxy whose ranking is known to be unreliable (Yu et al., 2020); independent retraining separates final evaluation from the shared-weight proxy, as in DARTS (Liu et al., 2019a).

D PROOFS

D.1 THE ADJACENT-LEVEL ESTIMATOR

Gradient estimation for discrete choices. Consider a discrete variable $\mathbf{z} \sim p_{\boldsymbol{\theta}}$ and an objective $f(\mathbf{z})$ that can be evaluated only at discrete values of $\mathbf{z}$; we follow the notation of the Monte Carlo gradient estimation literature (Mohamed et al., 2020), and Sec. 3.2 takes $\mathbf{z}$ to be the pre-acquisition rounding bits and f the validation cost. Since $\partial f / \partial \mathbf{z}$ does not exist, what can be optimized by gradient descent is the expected objective $\mathbb{E}_{p_{\boldsymbol{\theta}}}[f(\mathbf{z})]$, a smooth function of $\boldsymbol{\theta}$. The score-function estimator

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}}[f(\mathbf{z})] = \mathbb{E}_{p_{\boldsymbol{\theta}}}\left[f(\mathbf{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{z})\right] \tag{13}$$

is unbiased for any f (shown below) and never differentiates f, which is what makes it applicable; its price is a variance that can be large (Williams, 1992), since one sample says how good that draw was, not how it compares with the alternatives. Control variates subtract a baseline to reduce this variance (Tucker et al., 2017; Grathwohl et al., 2018). The pathwise estimators that NAS uses for the operation choice, straight-through and Gumbel-softmax (Jang et al., 2017; Maddison et al., 2017;

Liu et al., 2023), instead relax the variable to a continuous surrogate and differentiate f through it; this is biased and, more importantly here, requires f to be evaluable at a non-discrete input, which is unavailable when f is defined only at discrete points. Rao–Blackwellization lowers the variance of an unbiased estimator without introducing bias (Titsias & Lázaro-Gredilla, 2015; Liu et al., 2019b; Paulus et al., 2021): replacing the single-sample estimator by its conditional expectation given any statistic $T(\mathbf{z})$ of the sample leaves the expectation unchanged and, by the law of total variance, can only reduce the variance. For a factorized Bernoulli $\mathbf{z} = \mathbf{b} = (b_1, \ldots, b_Q)$, $b_q \sim \mathrm{Bernoulli}(\delta_q)$, conditioning coordinate q on the other bits $\mathbf{b}_{-q}$ gives

$$\mathbb{E}\left[f(\mathbf{b}) \, \frac{b_q - \delta_q}{\delta_q(1-\delta_q)} \;\middle|\; \mathbf{b}_{-q} \right] = f_q^+(\mathbf{b}_{-q}) - f_q^-(\mathbf{b}_{-q}), \tag{14}$$

where $f_q^{\pm}(\mathbf{b}_{-q})$ denotes f with b_q set to 1 or 0: the noisy score weight, which diverges as δ_q approaches 0 or 1, has disappeared, and what remains is the contrast between the objective at the two values of one bit with every other coordinate held fixed, still evaluated only at discrete points (derivation below). Sec. 3.2 uses Eq. 14 as the gradient of the pre-acquisition selector.

Unbiasedness of the score-function estimator (Eq. 13). For a discrete $\mathbf{z}$ with $J(\boldsymbol{\theta}) = \mathbb{E}_{p_{\boldsymbol{\theta}}}[f(\mathbf{z})] = \sum_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{z}) f(\mathbf{z})$ and f not depending on $\boldsymbol{\theta}$, the log-derivative identity $\nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{z})$ gives $\nabla_{\boldsymbol{\theta}} J = \sum_{\mathbf{z}} f(\mathbf{z}) \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbb{E}_{p_{\boldsymbol{\theta}}}\big[f(\mathbf{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{z}) \big]$, so the single-sample estimate $f(\mathbf{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{z})$ has expectation $\nabla_{\boldsymbol{\theta}} J$ exactly (Williams, 1992; Mohamed et al., 2020). The sum is finite here ($|\mathcal{S}|$ configurations), so no interchange of differentiation and integration needs justification.

Setup and the score weight. This is Eq. 14 above with $\mathbf{z} = \mathbf{b}$ and $f = F$. Let ξ collect the shared architecture-sampling noise and endpoint-specific mini-batch draws, independently of the rounding bits and selector parameters (with its dependence in F suppressed); unconditional expectations below also average over ξ. Fix an axis q and condition on $(\mathbf{b}_{-q}, \xi)$. For $0 < \delta_q < 1$, under $b_q \sim \mathrm{Bernoulli}(\delta_q)$ the log-likelihood of the axis is $b_q \log \delta_q + (1 - b_q) \log(1 - \delta_q)$, whose derivative with respect to δ_q is the score weight

$$w_q(b_q) = \frac{\partial}{\partial \delta_q} \log p_{\boldsymbol{\theta}}(b_q) = \frac{b_q - \delta_q}{\delta_q(1-\delta_q)}, \tag{15}$$

which takes the value $1/\delta_q$ at $b_q = 1$ and $-1/(1-\delta_q)$ at $b_q = 0$ and diverges as δ_q approaches either endpoint. The single-sample estimator of Eq. 13 for this coordinate is $F(\mathbf{b})\, w_q(b_q)$. Since F is a loss and therefore positive, this single-sample update always moves probability away from the level that was just drawn, by an amount proportional to F; which of the two levels is better is carried only by the difference between their costs, and the absolute value of F is noise. The conditional expectation below keeps exactly that difference.

Derivation of Eq. 14. Since b_q is Bernoulli, the conditional expectation of the single-sample estimator given $(\mathbf{b}_{-q}, \xi)$ is a sum of two terms, and the two score weights cancel against the two probabilities:

$$\mathbb{E}\big[F(\mathbf{b})\, w_q(b_q) \mid \mathbf{b}_{-q}, \xi \big] = \delta_q \, F_q^+(\mathbf{b}_{-q}) \frac{1}{\delta_q} + (1-\delta_q) \, F_q^-(\mathbf{b}_{-q}) \frac{-1}{1-\delta_q} = F_q^+(\mathbf{b}_{-q}) - F_q^-(\mathbf{b}_{-q}), \tag{16}$$

which is Eq. 14. Taking the outer expectation over $(\mathbf{b}_{-q}, \xi)$ and using the tower property leaves the expectation of the single-sample estimator unchanged, so the adjacent-level contrast is unbiased for $\partial\, \mathbb{E}_{\mathbf{b},\xi}[F(\mathbf{b})] / \partial \delta_q$. The same result follows directly: since b_q is independent of $\mathbf{b}_{-q}$, the expected cost is affine in δ_q, $J = \mathbb{E}_{\mathbf{b}_{-q},\xi}\big[\delta_q F_q^+(\mathbf{b}_{-q}) + (1-\delta_q) F_q^-(\mathbf{b}_{-q}) \big]$, whose derivative in δ_q is $\mathbb{E}_{\mathbf{b}_{-q},\xi}[F_q^+ - F_q^-]$; the gradient with respect to the rounding probability of one axis is exactly the expected difference between the costs at its two adjacent levels.

Variance. By the law of total variance, for any statistic T and square-integrable integrand G, $\mathrm{Var}[G] = \mathbb{E}\big[\mathrm{Var}[G \mid T]\big] + \mathrm{Var}\big[\mathbb{E}[G \mid T]\big] \geq \mathrm{Var}\big[\mathbb{E}[G \mid T]\big]$. With $G = F(\mathbf{b})\, w_q(b_q)$ and $T = (\mathbf{b}_{-q}, \xi)$ the right-hand side is the variance of the adjacent-level contrast, so the contrast never has larger variance than the single-sample score estimator, and the gap is the expected conditional variance that the divergent weight of Eq. 15 contributes. In the terms of the Rao–Blackwell construction, the conditioning statistic is $T = (\mathbf{b}_{-q}, \xi)$, the randomness averaged out is the single bit b_q, and the conditional expectation is computed exactly rather than estimated, because b_q has only

two outcomes and both endpoint costs are measured; this is the local expectation gradient of Titsias & Lázaro-Gredilla (2015) for a binary coordinate, where the sum over the K values of a categorical coordinate reduces to two terms. The randomness that remains is $\mathbf{b}_{-q}$ together with the mini-batch and architecture-sampling noise inside F; the latter is not marginalized: architecture-sampling noise is shared across endpoints, while mini-batches are drawn separately. This sharing preserves endpoint marginals and is separate from the Rao–Blackwell step.

Proposition 1 (Unbiasedness and variance of the adjacent-level contrast)**.** *Fix W, Ψ, ω and let θ_q lie inside an active interval. Then the stochastic gradient of $\mathcal{L}_{\mathrm{cfg}}$ (Eq. 5) is an unbiased estimator of $\partial J/\partial\theta_q$ in Eq. 6, and its variance does not exceed that of the single-sample score-function estimator of the same gradient.*

Proof. Within the active interval j_q is constant and $\partial\delta_q/\partial\theta_q = (K_q-1)m_q(1-m_q)$ is deterministic, so both estimators are the same factor times their δ_q counterparts. Unbiasedness follows from Eq. 16 and the tower property, and the variance bound from the law of total variance with $T = (\mathbf{b}_{-q}, \xi)$, as derived above. □

Chain rule to θ_q. Inside the active cell the integer part j_q of Eq. 3 is constant, so with $m_q = \mathrm{sigm}(\theta_q)$, $r_q = (K_q - 1)m_q$ and $\delta_q = r_q - j_q$,

$$\frac{\partial\delta_q}{\partial\theta_q} = \frac{\partial r_q}{\partial\theta_q} = (K_q - 1)\,\frac{\partial\,\mathrm{sigm}(\theta_q)}{\partial\theta_q} = (K_q - 1)\,m_q(1 - m_q), \tag{17}$$

the sigmoid derivative scaled by the number of cell widths on the axis. Multiplying Eq. 16 by Eq. 17 gives Eq. 6. At an interior grid boundary, the floor convention selects the right-hand interval; the update uses its one-sided derivative, without assuming differentiability across the boundary.

Base-and-flip with common random numbers. One base configuration $\mathbf{b}$ is drawn from $p_{\boldsymbol{\theta}}$ and F is evaluated at it. For each axis q, we flip b_q while keeping $\mathbf{b}_{-q}$, W, Ψ, ω, and the shared Gumbel noise fixed. At the flipped configuration, we recompute the representation and architecture logits before sampling an architecture and evaluating F. The base evaluation supplies one endpoint of every axis and each flip supplies the other, so $Q+1$ evaluations of F per step deliver all Q contrasts (four on radar, where $Q = 3$: the base and its three single-axis neighbours); each evaluation is a forward pass on B validation mini-batches with no backward pass. The single-sample score-function alternative is ablated as w/o RB in Sec. 4.4; categorical selection mechanisms for the acquisition axes are compared in Sec. 4.3. Sharing random numbers preserves the endpoint marginals and can reduce the variance of their difference through positive covariance, but does not guarantee a reduction. This coupling is separate from Rao–Blackwellization.

D.2 Identifiability of the structured representation

Informally, after centering, no part of the interaction can be rewritten as a baseline or a per-axis effect, so the three parts of Eq. 7 are uniquely determined.

Proposition 2 (Identifiability of the structured representation)**.** *For each interaction table (the three radar axes, Eq. 11; each IMU row–column table, Eq. 12), let $\mathbf{R}^{\mathrm{eff}}$ be obtained by removing every one-axis marginal. Then every one-axis marginal of $\mathbf{R}^{\mathrm{eff}}$ is zero, interactions of two or more axes are preserved, and the map $\mathbf{R}^{\mathrm{raw}} \mapsto \mathbf{R}^{\mathrm{eff}}$ is idempotent. Thus, for a fixed representation function, the baseline, centered axis effects, and interaction of Eq. 7 are uniquely determined under these centering constraints. This does not imply identifiability of the raw parameters or of the hypernetwork.*

Proof sketch. Let n be the number of axes of the table (3 on radar, 2 on IMU). Averaging $\mathbf{R}^{\mathrm{eff}}_{\mathbf{s}}$ over all axes except q gives, with $\overline{\mathbf{R}}^{\mathrm{raw}}_{q,s_q}$ and $\overline{\mathbf{R}}^{\mathrm{raw}}$ as in Eq. 11, $\overline{\mathbf{R}}^{\mathrm{raw}}_{q,s_q} - \overline{\mathbf{R}}^{\mathrm{raw}}_{q,s_q} - (n-1)\overline{\mathbf{R}}^{\mathrm{raw}} + (n-1)\overline{\mathbf{R}}^{\mathrm{raw}} = 0$, so no component of $\mathbf{R}^{\mathrm{eff}}$ can be absorbed by $\boldsymbol{\mu}$ or by an axis table, whose entries are functions of one s_q; the subtracted terms are themselves functions of one axis or constants, so any component depending jointly on two or more axes passes unchanged. Applying the map twice subtracts the one-axis marginals of $\mathbf{R}^{\mathrm{eff}}$, which are zero, hence idempotence. Centering the axis tables removes their means, which are absorbed by $\boldsymbol{\mu}$. □

E EXPERIMENTAL SETUP

Radar. Five participants (P1–P5) performed the seven marshalling gestures of Müller et al. (2023) (emergency stop, move ahead, move back, slow down, start engine, stop engine, and a no-gesture class) in front of an FMCW radar under all 27 combinations of ADC samples per chirp $\{64, 160, 256\}$, chirps per frame $\{64, 160, 255\}$, and chirp slope $\{15, 30, 45\}$ MHz/μs. Every configuration is a separately recorded data set, so the corpus consists of 135 participant–configuration recordings rather than one recording resampled offline. Each participant's 27 configurations were recorded independently across several sessions on separate days. Configurations were recorded as separate takes with rest between them and never continuously with one another, so trials carry no temporal continuity across takes or configurations and recording order does not correlate with a configuration's index. Recordings were collected at the same location and under the same experimental environment across participants and acquisition configurations. Each frame is turned into a micro-Doppler spectrogram that the network consumes. The common radar protocol retrains a network from scratch on four participants for 50 epochs (Adam, learning rate 10^{-3}, weight decay 10^{-4}, cosine decay to zero, last epoch reported) and reports the mean test accuracy over three participant-held-out folds, each testing the held-out participant (Participant splits below).

IMU. The garment benchmark of Tsukamoto et al. (2023) provides 396 inertial-sensor locations and 14 activities (walking, jogging, going up and down stairs, standing, sitting, typing, writing, eating a cookie, eating pasta, ironing, folding, tooth brushing, and vacuuming) from 12 participants; participants 1–9 train, participant 10 validates, and participants 11–12 are held out for scoring. Three sensor slots are assigned to the front-upper panel and one to the front-lower panel; each slot carries the ordinal row and column coordinate within its panel. Slots that select the same sensor are merged, so a deployed placement may use fewer than four sensors. Grid positions without a recorded sensor map to the last recorded sensor in the same row. The common protocol trains for 150 epochs with the same optimizer and schedule and reports macro-F1 on the two held-out participants.

Participant splits. On radar, search uses a fixed pool of four participants (P1–P4); Phase A and the black-box campaigns use within-participant training/validation splits in that pool, and Phase B trains on P1–P3 and validates on P4, with its internal two-participant candidate checks on P4 and P2. Final scoring is separate from search: every reported candidate is retrained from scratch on four participants and tested on the held-out fifth, with P1, P3, and P5 held out in turn, and the three accuracies are averaged; P5 is never used in search. On IMU, search and retraining use participants 1–9 for training and participant 10 for validation, and participants 11–12 are used only for the final test. No weights from search enter a scored network, and the same splits score every method. On radar, to check generalization to a participant unseen by the search, Table 8 recomputes HV and best accuracy for the radar fronts of Table 1 with each point scored on the two held-out participants that contributed to search (P1 and P3, mean of their folds) and, separately, on the participant excluded from search (P5): SPANAS ranks first in both cases, and the strongest campaign is the same in both.

Table 8: Radar fronts of Table 1 rescored per held-out participant group (mean over three searches): HV and best accuracy on the two held-out participants that contributed to search (P1, P3) and on the participant never used in search (P5).

	P1, P3 (used in search)		P5 (never used in search)	
Method	HV↑	Best acc.↑	HV↑	Best acc.↑
qNParEGO	0.302	0.872	0.327	0.922
qNEHVI	0.285	0.846	0.213	0.689
NSGA-II	0.261	0.818	0.247	0.833
Random search	0.302	0.862	0.310	0.870
SPANAS (ours)	**0.334**	**0.897**	**0.374**	**0.948**

Baseline protocols. Table 1 compares SPANAS with four black-box campaigns that jointly search acquisition settings and architectures, each with $N{=}100$ candidate evaluations. The NAS baselines of Table 3 are described in Appendix G. Table 2 reports domain-specific models (Mutegeki et al., 2021; Lu et al., 2022; Ordóñez & Roggen, 2016) at the IMU placement recommended by Tsukamoto et al. (2023) (68, 90, 108, 241) under the common protocol. The IMU cell operations (none, skip,

convolutions with kernel sizes 1, 3, and 5, pooling, a GRU, and attention) include the convolutional and recurrent operator families these models are built from, so the joint search space covers their building blocks. The architecture-only control applies SPANAS refinement and verification at that same fixed placement, without acquisition search; its four verified cells, 000200, 215222, 315222, and 415222, appear in Table 2 in increasing storage order. The SPANAS (ours) rows are the non-dominated candidates pooled over the three joint searches of Table 1, each at its searched placement; pooled rows are individual verified candidates, not averages over searches. The NAS extensions to the acquisition axes (Sec. 4.3) run inside the SPANAS pipeline with the selector replaced and are distinct from these architecture-search baselines. The number of distinct candidates produced by refinement can be smaller than the cap k. In Table 4, the radar Full row reuses the refinement candidate set and verified scores of one of the three searches of Table 1; the IMU Full row is a separate run of the full pipeline under the ablation settings, with 16 verified candidates, whose candidates therefore do not appear in Tables 2, 3, and 11. The radar variants run the full pipeline: w/o ordinal structure selects 160/160/30, whereas w/o RB and w/o centering and projection select 160/255/30 and, because Phase B replaces the representation by a free logit table, reproduce the candidate set of Full. Each domain's Full and variants use the same domain-specific settings except for the specified component, including search budget, refinement, and final scoring protocol. The radar w/o structured representation (free lookup) row runs the same pipeline with the representation replaced by a free lookup table; its refinement yields three verified candidates. The IMU entry is not applicable: a free lookup table needs one vector per configuration, and the IMU space has 396^4 placements.

Pipeline settings. Each reported search runs Phase B once at its selected configuration and verifies every candidate the refinement produces. Table 9 lists the settings of both domains. In both, the structured representation has dimension 20 and the hypernetwork one hidden layer of width 64. On radar, the Phase-A cell reported in Table 4 (w/o Phases B and C) is decoded after $2{,}000$ further steps of shared-weight and generator updates at the selected configuration; refinement re-initializes the supernetwork, so these steps do not affect the other rows.

Table 9: Pipeline settings per domain.

Setting	Radar	IMU
Phase A: search steps (of which warm-up)	10,000 (7,500)	24,000 (2,000)
Phase A: validation mini-batches B per evaluation of F	7	8
Phase A: Gumbel-softmax temperature τ	1	1
Phase A: selector learning rate (Adam)	0.06	0.06
Storage coefficient λ, Phase A / Phase B	0.02 / 0	0.01 / 0.01
Phase B: uniform steps	7,500	12,000
Phase B: commitment rounds $\times$ steps	$6 \times 1{,}500$	$6 \times 4{,}800$
Phase B: tail steps	4,000	8,000
Phase B: architecture-sampling temperature	$1.0 \rightarrow 0.3$	$1.0 \rightarrow 0.3$
Phase B: candidate cap k	20	20

Campaigns. qNEHVI and qNParEGO use the BoTorch implementations (Balandat et al., 2020) with a Sobol initial design (qNParEGO is BoTorch's parallel, noisy variant of ParEGO (Knowles, 2006), with random augmented Chebyshev scalarizations), NSGA-II the pymoo implementation (Blank & Deb, 2020), and random search draws configurations and cells uniformly; all place the acquisition variables and the six cell operations in one decision vector and run for 100 candidate evaluations, each candidate trained from scratch on the training participants at search time. The front points of every campaign are then retrained under the common protocol.

Metrics. Table 1 reports means and sample standard deviations over three searches. For each run, "Best" is the maximum test score among independently verified candidates. Hypervolume, GD, and IGD are computed in a normalized objective space (storage divided by 5 056 KB on radar and 1 750 KB on IMU; $1-$accuracy or $1-$macro-F1) with reference point $(1.05, 0.45)$ on radar and $(1.05, 0.40)$ on IMU; the reference front of a search is the non-dominated union of the fronts of all methods in that search. For Table 4, the reference front is the non-dominated union of the campaign fronts and the SPANAS front of the same search; all radar rows except the free lookup, which comes from a separate search and is scored against that search's reference front, use the same search.

Storage is that of the deployed subnetwork, counting only its reachable operations: fp32 parameters on radar, and on IMU the benchmark's deployment convention of 8-bit convolutional and recurrent weights with fp32 linear and normalization layers, computed by one function for every searched cell and every domain-specific model in Table 2.

Hardware. Experiments were run on NVIDIA RTX A5000, RTX A6000, RTX 3090, and RTX 5090 GPUs.

F FULL RESULT TABLES

Tables 10 and 11 list every non-dominated operating point of every method under the common protocol, pooled over the three searches; the per-search metrics of Table 1 are computed against the reference front of each search.

Table 10: Radar non-dominated operating points, pooled over the three searches under the common protocol (Table 1). Bold highlights SPANAS and the storage/accuracy extrema.

		Acquisition configuration				
Method	Cell	ADC samples	Chirps	Slope (MHz/μs)	Storage (KB)	Accuracy
qNParEGO (Knowles, 2006)	000400	256	160	45	**284**	0.619
	400120	256	160	15	394	0.689
	040202	160	255	30	503	0.864
	400030	160	255	30	1234	0.913
qNEHVI (Daulton et al., 2021)	000400	160	255	30	**284**	0.787
	204414	160	255	30	394	0.868
	010304	160	255	30	1234	0.879
NSGA-II (Deb et al., 2002)	000400	160	255	30	**284**	0.787
	400240	160	255	30	394	0.846
	300440	160	255	30	1234	0.870
Random search (Bergstra & Bengio, 2012)	000400	256	255	15	**284**	0.671
	040002	160	255	30	394	0.875
SPANAS (ours)	**040004**	**160**	**255**	**30**	**284**	**0.789**
	040002	**160**	**255**	**30**	**394**	**0.867**
	040003	**160**	**255**	**30**	**1234**	**0.913**
	030002	**160**	**255**	**30**	**1343**	**0.918**

Table 11: IMU non-dominated operating points, pooled over the three searches. Sensor indices denote recorded sites. Protocol: Appendix E. Bold highlights SPANAS and the storage/macro-F1 extrema.

		Acquisition configuration		
Method	Cell	Sensor sites	Storage (KB)	Macro-F1
qNParEGO (Knowles, 2006)	000500	29,115,292	**13**	0.675
	000500	65,113,115,278	44	0.758
	000300	65,67,76,278	210	0.761
	353425	65,88,285	345	0.765
	000300	65,82,86,287	375	0.822
qNEHVI (Daulton et al., 2021)	000500	35,64,83,252	**13**	0.788
	000400	63,82,91,229	210	0.817
	323545	90,91,107,254	345	**0.880**
NSGA-II (Deb et al., 2002)	010201	62,88,91,276	44	0.761
	020205	62,88,91,276	74	0.791
Random search (Bergstra & Bengio, 2012)	155115	53,61,115,298	**13**	0.733
	415051	49,67,115,300	149	0.796
	010401	62,79,115,277	232	0.865
SPANAS (ours)	**000200**	**43,71,107,253**	**44**	**0.757**
	010003	**12,72,111,288**	**96**	**0.822**
	000300	**12,72,111,288**	**127**	**0.823**
	010303	**12,72,111,288**	**179**	**0.842**
	040503	**12,72,111,288**	**232**	**0.863**

F.1 HYPERVOLUME PROGRESSION DETAILS

Figure 6 resolves the three runs averaged in Figure 4. Line styles identify the first, second, and third runs within each method; search seed identifiers are 0–2 for the campaigns and radar SPANAS, and 30–32 for IMU SPANAS. Campaign curves reconstruct validation-proxy histories calibrated to each run's verified endpoint; SPANAS adds independently verified candidates in archive order. Specifically, the proxy HV history of each campaign run is rescaled by a constant so that its value at evaluation 100 equals that run's verified HV in Table 1; intermediate values are therefore proxy-shaped rather than verified. Main curves average these three histories, and Table 1 reports the sample standard deviations of their verified endpoint metrics. Measured search times are given in Appendix F.2.

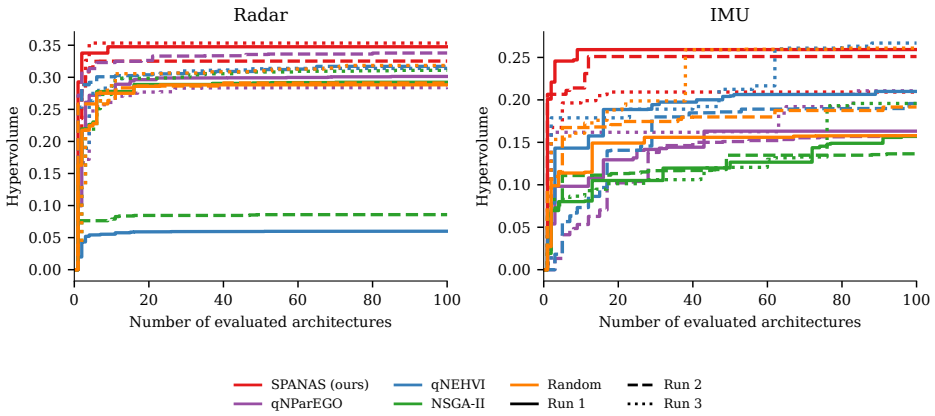


Figure 6: Individual-run HV histories underlying the main mean curves. Colors identify methods and line styles identify runs.

F.2 MEASURED SEARCH TIME

Table 12 lists measured wall-clock search times next to the mean HV of Table 1. Campaign times are the logged wall time of the 100 candidate evaluations of each search, averaged over the three searches; SPANAS times are the logged wall time of one search per domain through Phases A, B, and C, reported both up to the verified candidate at which the mean HV curve passes the final HV of the strongest campaign (three on radar, eleven on IMU; Fig. 4) and for all verified candidates. On radar, whose 27 configurations are visited often during Phase A, the selector converges early and SPANAS reaches a higher HV than every campaign in a comparable total time; on IMU, whose placement space is far larger, SPANAS spends more time than qNParEGO and qNEHVI and less than NSGA-II and random search, and reaches the highest HV.

Table 12: Measured wall-clock search time (hours) and mean HV. Campaigns: mean over three searches of 100 evaluations (HV of Table 1). SPANAS: one search per domain, up to the candidate at which the mean HV curve of Fig. 4 passes the strongest campaign, and for all verified candidates.

	Radar		IMU	
Method	Time (h)	HV	Time (h)	HV
qNParEGO	5.3	0.308	4.0	0.177
qNEHVI	4.6	0.230	4.1	0.224
NSGA-II	5.4	0.230	7.6	0.163
Random search	4.1	0.299	9.6	0.203
SPANAS, up to passing the strongest campaign	2.2	**0.324**	5.9	**0.230**
SPANAS, all verified candidates	4.3	**0.342**	6.5	**0.240**

G NAS BASELINES: EXTENSIONS TO THE ACQUISITION AXES AND BLACK-BOX-SELECTED SETTINGS

NAS baselines. Table 3 uses four differentiable NAS methods, which differ in how they handle a discrete choice: Gumbel-ST samples one option through a straight-through Gumbel-softmax relaxation; the DARTS mixture relaxes the choice to a softmax-weighted mixture of all options; ProxylessNAS samples two options at a time and trains their binary gates; DrNAS treats the option weights as a Dirichlet distribution and samples them. We apply each method in two ways.

NAS extended to the acquisition axes. The upper block of Table 3 keeps the SPANAS pipeline (configuration-conditioned generator, staged refinement, verification) but replaces the ordinal selector by each method's own choice mechanism, applied to the levels of every acquisition axis as unordered categories. Both domains report the best verified candidate of the full staged pipeline.

Black-box-selected settings followed by NAS. The lower block fixes the acquisition setting first and searches only the architecture. Settings are taken from the validation–storage Pareto front pooled over the four black-box campaigns and the three searches of each domain, excluding SPANAS candidates, and ranked on validation data before final scoring; this validation front differs from the verified fronts of Tables 10 and 11. Radar yields a single eligible setting, 160/255/30; on IMU the three highest-ranked placements are 26,86,113,305; 65,113,115,278; and 29,115,292. Each method searches a cell at these settings with the same budget (50 epochs, batch size 32) and decoding, without progressive refinement or a shortlist, and the decoded cell is independently retrained; Table 3 reports the best of the three IMU settings.

H REPRODUCIBILITY DETAILS

Every reported number is produced by a single deterministic run with training seed 0: cuDNN kernels are deterministic, TF32 is disabled, the cuBLAS workspace is fixed, and data loaders are seeded. The three searches of Sec. 4 differ only in the randomness of the search and refinement stages; the common-protocol training that scores every front point is identical across them. The exact settings and commands for the main experiments (Table 1) will be released with the code.