PHASE-FREE BIDIRECTIONAL STRUCTURAL ADAPTATION IN BAYESIAN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological neural systems continuously adapt their structure through the formation and elimination of neural connections, whereas artificial neural networks typically retain a fixed architecture throughout training. Amongst existing structural adaptation approaches, most rely on pruning rather than growth, with the combination of the two being the exception, typically relying on pre-defined adaptation phases. We propose, to the best of our knowledge, the first phase-free bidirectional structural adaptation algorithm that leverages parameter statistics in a Bayesian neural network framework to control optional growth or pruning steps through a single unified criterion. Concretely, candidates for growth and pruning are proposed based on weight-level uncertainty and signal-to-noise-ratio, while candidate selection follows a common penalised evidence lower bound that explicitly accounts for model complexity. This enables network capacity to continuously increase, decrease, or remain unchanged throughout training. We evaluate the proposed framework on fully connected and convolutional architectures using Fashion-MNIST and CIFAR-10. Across our experiments, the method achieves favourable performance–complexity trade-offs compared with static architectures and dynamic baselines that rely on predefined growth and pruning schedules or separate structural criteria.

1 INTRODUCTION

Biological neural systems exhibit substantial structural adaptation through mechanisms including synaptic formation (or neurogenesis) and pruning (or apoptosis), often referred to as neuroplasticity (Raff et al., 1993; Götz & Huttner, 2005). Artificial neural networks (ANNs) in the deep learning (DL) era, however, mostly take a fixed and pre-defined structure, which does not change during training. Traditionally, determining the model's capacity and, therefore, the corresponding hypothesis space is closely related to effects like overfitting and underfitting. Practical solutions to find an appropriate neural network size can be performed through neural architecture search (NAS), during which models are commonly trained repeatedly from scratch, with candidate architectures being explored through grid search, random search, or evolutionary and genetic algorithms (Bergstra & Bengio, 2012; Elsken et al., 2019; Zhang et al., 2018). These training paradigms require notable computational overhead and rely on static network architectures.

However, several approaches have highlighted the possible benefits of architecture adjustments during training: Most famously, dropout (Hinton et al., 2012; Srivastava et al., 2014) and dropconnect (Wan et al., 2013) randomly deactivate neurons and connections during training only to discover more favourable minima in the loss landscape, while pruning (LeCun et al., 1989) and structural pruning (Wen et al., 2016) seek to eliminate connections and neurons entirely, mostly to reduce computational costs while keeping performance high. In contrast, mechanisms that allow networks to introduce capacity dynamically during training are sparse and have not reached the status of universal appreciation as the aforementioned capacity reduction techniques (Huang et al., 2023; Sowrirajan et al., 2024).

Even less frequent are approaches which play with both increasing and decreasing capacity simultaneously. Amongst them, structural adaptation is typically constrained by predefined phases (De Lambilly & Duffner, 2026; Li et al., 2026), which ultimately does not allow the network to discover an optimal capacity for the task at hand. Instead, their benefit likely lies in a

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.

wider exploration of the loss landscape during optimisation. NeST (Dai et al., 2019) is one of the few examples that allows a model to dynamically grow and shrink capacity while relying on two distinct criteria: gradient signals for growing and weight magnitude for pruning. However, all capacity-controlling approaches in the current literature split growing and pruning processes into distinct phases, requiring a high level of manual control, which allow for a phase-free process to dynamically adjust network capacity.

In this work, we aim to further explore the current frontiers by defining for the first time a single, unified, statistical criterion that allows us to control both aspects of neuroplasticity in a phase-free optimisation algorithm. We thereby tackle one of the biggest challenges related to neuroplastic artificial neural networkss (ANNs), which is to define appropriate criteria for capacity growth and shrinkage (Li et al., 2025), the latter being referred to as pruning. We argue that Bayesian neural networkss (BNNs) (Blundell et al., 2015) leverage particular advantages to define such criteria, as their probabilistic nature provides an intuitive approach to quantify network characteristics relevant to add or remove capacity. Concretely, our suggested approach relies on Bayesian posterior uncertainty to guide where growth is proposed, while posterior signal-to-noise ratio (SNR) identifies pruning candidates. Growth and pruning refer in this context to the introduction and removal of individual (or a certain percentage) of neurons in the case of fully connected layers or filters in the case of convolutional neural networkss (CNNs). Crucially, these statistics are used only to construct candidate architectures with increased, decreased and identical capacity, and the final structural decision is made by comparing all candidates under a common *penalised evidence lower bound (ELBO)*, which augments the variational objective with an explicit parameter-count penalty.

Our contributions are as follows:

- We propose, to the best of the authors' knowledge, the first neuroplasticity algorithm for BNNs, which relies only on a single unified criterion to allow both grow and prune neural architectures dynamically in a phase-free manner.
- We demonstrate performance and efficiency benefits of our suggested approach for both fully connected and convolutional architectures on the standard vision benchmark datasets FashionMNIST and CIFAR10, comparing to both state-of-the-art approaches and static ANN architectures.
- We run ablation studies to show how certain design choices impact the algorithm's behaviour and analyse the neuroplastic behaviour of the algorithm in various settings.

2 Related Work

2.1 Structural Contraction: Network Pruning

A substantial body of research has investigated network compression by removing redundant parameters from over-parameterised models. At the weight level, Han et al. (2015) introduced iterative magnitude pruning, which adopts a three-stage procedure consisting of training, removing low-magnitude weights, and subsequent fine-tuning, which achieved a nine-fold reduction in the number of parameters. However, weight-level pruning typically produces unstructured sparsity, resulting in irregular connectivity patterns. Li et al. (2017) instead identify and remove filters that have limited impact on the output accuracy, while Molchanov et al. (2019) estimate the contribution of individual filters to the final loss using first- and second-order Taylor expansions. Bayesian approaches further formulate pruning as a probabilistic inference problem, replacing heuristic importance criteria with posterior statistics. Louizos et al. (2017) introduce hierarchical priors that enable the pruning of entire nodes rather than individual weights, whereas Neklyudov et al. (2017) employ a truncated log-normal noise model to identify and remove parameters with low signal-to-noise ratios from the computational graph. Despite their methodological differences, these approaches are inherently irreversible. Once a component is pruned, the lost capacity cannot be recovered. Moreover, existing pruning criteria are primarily used to identify components for removal, rather than to guide the dynamic adaptation of network capacity. Our framework instead reuses posterior mean and variance to guide both pruning and expansion.

2.2 Structural Expansion: Network Growth

While pruning methods focus on reducing network capacity, a complementary line of work considers the opposite problem that increasing capacity when the current architecture is insufficient. Function-preserving expansion was pioneered by Net2Net (Chen et al., 2016), which introduces function-preserving transformations to initialise deeper networks while preserving the function of the original model. Wu et al. (2020) generalise this idea into a framework for progressively and dynamically growing neural networks, identifying local expansion opportunities through Taylor approximation, while GradMax (Evci et al., 2022) initialises new neurons to maximise the gradient of the loss with respect to the added weights. Task-driven approaches provide another mechanism for dynamically expanding network capacity. Progressive Networks add a new column for each task while freezing previously learnt parameters (Rusu et al., 2016), whereas DEN dynamically adapts network capacity through selective retraining and neuron splitting (Yoon et al., 2018). Bayesian nonparametric approaches instead treat model capacity as a latent variable. The Indian Buffet Process (IBP) places a prior over an unbounded number of latent features (Ghahramani & Griffiths, 2005), while stick-breaking constructions enable deep models to infer their effective width automatically (Nalisnick & Smyth, 2017). Despite these advances, pruning and expansion remain largely separate, with different criteria guiding capacity changes. Our framework addresses this limitation by reusing posterior statistics for both growth and pruning, with a unified penalised ELBO providing a consistent criterion for capacity adaptation.

2.3 Joint Growth-and-Pruning

The closest line of work to ours jointly performs structural growth and pruning within a single training run. NeST (Dai et al., 2019) iteratively grows and prunes neurons and connections using gradient- and magnitude-based criteria. Dynamic sparse training methods such as SET (Mocanu et al., 2018) and RigL (Evci et al., 2020) maintain a fixed parameter budget while rewiring the network, using weight magnitude for pruning and gradient information for growth; subsequent methods further refine these criteria or update schedules (Mostafa & Wang, 2019; Jayakumar et al., 2020; Liu et al., 2021). However, these approaches typically treat growth and pruning as separate operations, relying on heterogeneous criteria and an implicitly fixed parameter budget. Such heterogeneous criteria make growth and pruning difficult to compare under a common objective, leaving no principled basis for deciding whether capacity should be added, removed, or retained. Our framework instead evaluates all three choices under a single penalised ELBO, where posterior statistics guide both structural changes and the associated capacity cost.

2.4 Bayesian Neural Networks & Variational Inference

Addressing all three limitations simultaneously requires a framework that compares grow, prune, and retain candidates on a common numeric scale at each decision point, derives principled importance measures for both directions of structural change from the same underlying statistics, and incorporates model complexity directly into the training objective. BNNs provide a natural substrate for satisfying these requirements: variational inference yields per-parameter posterior distributions whose means and variances can be used to derive both signal-to-noise-based pruning criteria and uncertainty-based growth criteria, while the ELBO explicitly balances data fit against model complexity. We therefore review the BNN and variational inference formulation.

Graves (2011) provided an early practical recipe for VI in differentiable networks, showing that the resulting posterior can support effective pruning. Blundell et al. (2015) subsequently introduced Bayes by Backprop for scalable variational learning of weight distributions, while Kingma et al. (2015) proposed local reparameterization to reduce gradient variance. Importantly, Bayesian model evidence naturally incorporates a complexity penalty. MacKay (1992) showed that it embodies an Occam-style preference for simpler models and provides principled criteria for network pruning and growth. Posterior statistics such as signal-to-noise ratios have subsequently been used for pruning (Neklyudov et al., 2017). However, existing approaches do not, to our knowledge, jointly evaluate growth, pruning, and retention using shared posterior statistics and a common complexity-aware objective. Our penalised ELBO provides such a unified criterion, using posterior mean and variance to guide growth and pruning while allowing all three actions to be evaluated through a single objective.

Algorithm 1 Phase-free structural adaptation

Require: $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, $(A^0, \boldsymbol{\phi}^0)$, T, M, M_1, K, ρ, β, λ
for $t = 0, \ldots, T-1$ **do**
  $\boldsymbol{\phi}_{\text{cur}} \leftarrow \text{WeightOptimise}(A^t, \boldsymbol{\phi}^t, \mathcal{D}_{\text{train}})$
  $(A^+, \boldsymbol{\phi}_0^+) \leftarrow \text{Grow}(A^t, \boldsymbol{\phi}_{\text{cur}})$; $(A^-, \boldsymbol{\phi}_0^-) \leftarrow \text{Prune}(A^t, \boldsymbol{\phi}_{\text{cur}}, \rho)$
  Form $\mathcal{C} = \{(A^t, \boldsymbol{\phi}_{\text{cur}}), (A^+, \boldsymbol{\phi}_0^+), (A^-, \boldsymbol{\phi}_0^-)\}$
  Sample shared training and validation mini-batches
  for each $(A', \boldsymbol{\phi}_0') \in \mathcal{C}$ **do**
    $\boldsymbol{\phi}_{A'} \leftarrow \text{CandidateTrain}(A', \boldsymbol{\phi}_0', M, M_1)$
    Evaluate $s(A')$ using equation 5
  end for
  $(A^{t+1}, \boldsymbol{\phi}^{t+1}) \leftarrow \arg\max_{(A', \boldsymbol{\phi}_{A'}) \in \mathcal{C}} s(A')$
end for
return $(A^T, \boldsymbol{\phi}^T)$

3 METHOD

Addressing the absence of a principled, unified objective governing both pruning and growth in prior work, we propose a methodology that dynamically adapts the width of a BNN during training. Concretely, we operate at the level of individual structural units (SUs) within a layer. While our primary focus is on fully connected BNNs, where an SU corresponds to a hidden neuron, the same formulation extends naturally to convolutional networks, where an SU corresponds to a filter. At structural decision time t, the current architecture A^t gives rise to three candidate changes: grow by one SU, prune a fraction of the least informative SUs, or retain the current architecture. The pipeline is shown in Figure 1, where growth and pruning remain available throughout training rather than being confined to separate phases.

3.1 A COMMON OBJECTIVE FOR STRUCTURAL DECISIONS

Each architecture A is paired with a mean-field variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A)$, trained by Bayes by Backprop (Blundell et al., 2015). We compare architectures using

$$\mathcal{J}(A, \boldsymbol{\phi}; \mathcal{B}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p(\mathcal{B}_y \mid \mathcal{B}_x, \boldsymbol{\theta}, A)] - \beta D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \,\|\, p(\boldsymbol{\theta} \mid A))}_{\mathcal{L}_\beta(A, \boldsymbol{\phi}; \mathcal{B})} - \lambda N(A), \tag{1}$$

where $\mathcal{B}$ is a mini-batch, $N(A)$ is the number of parameters, $\beta > 0$ weights the parameter-level KL term, and $\lambda \geq 0$ controls the preference for compact architectures. For $\beta = 1$, $\mathcal{L}_\beta$ is the usual evidence lower bound (ELBO); other values give a KL-weighted variational objective. These posterior statistics are used to generate the pruning and growth candidates, while the decision of which candidate to adopt is based on independently training all candidates for the same short horizon and comparing their performance on shared validation batches. The explicit $\lambda N(A)$ term makes the predictive-fit–complexity trade-off comparable across candidates, in the spirit of evidence-based Bayesian model selection (MacKay, 1992). Appendix A gives the corresponding joint variational interpretation over weights and architectures. The two complexity terms play different roles: the weight-level KL regularises the posterior within a fixed architecture, whereas $\lambda N(A)$ assigns an explicit cost to changing the architecture itself. Including A^t among the candidates is equally important, since a structural update is made only when growth or pruning improves on continued training of the current model. The methodology of growth and pruning are detailed as follows.

3.2 CONSTRUCTING STRUCTURAL CANDIDATES

Growth. The "growth" targets the layer whose posterior-variance distribution has the strongest upper tail relative to its own scale. For adaptable layer l, let $\mathcal{W}^{(l)}$ contain the weights incident to its units, including incoming and outgoing connections. We standardise each posterior variance using the within-layer median and median absolute deviation (MAD). This relative score avoids an

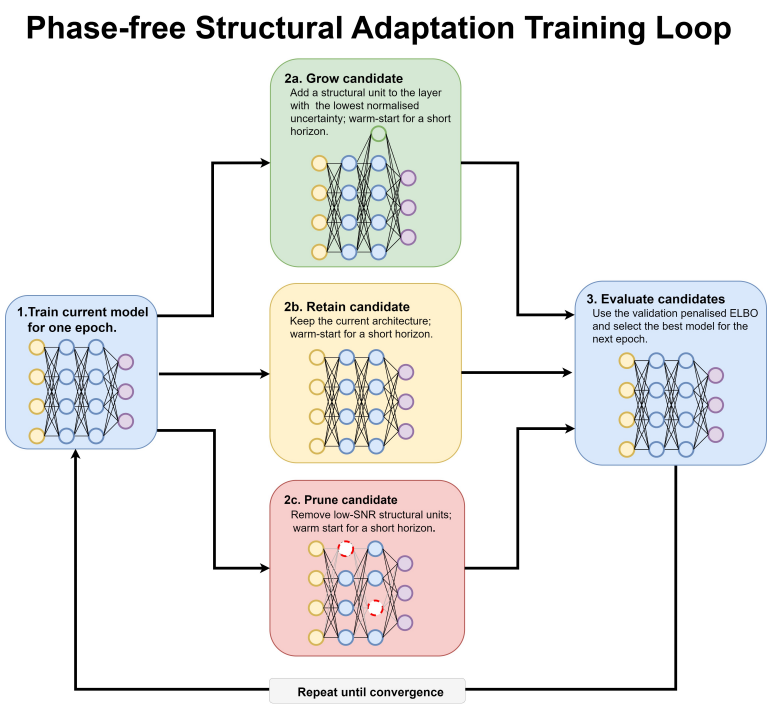


Figure 1: One structural decision. After updating the current BNN, we form grow, retain, and prune candidates, train each on the same mini-batches, and carry forward the candidate with the largest validation score.

absolute threshold, which could systematically favour layers with larger posterior scales:

$$m^{(l)} = \underset{w\in\mathcal{W}^{(l)}}{\text{median}}\,\sigma_w^2, \qquad d^{(l)} = \underset{w\in\mathcal{W}^{(l)}}{\text{median}}\,|\sigma_w^2 - m^{(l)}|,$$
$$z_w^{(l)} = \frac{\sigma_w^2 - m^{(l)}}{d^{(l)} + \epsilon}, \qquad \mathcal{U}^{(l)} = Q_{0.9}\Big(\{z_w^{(l)} : w \in \mathcal{W}^{(l)}\}\Big), \tag{2}$$

where $\epsilon > 0$ prevents division by zero. MAD scaling limits the influence of extreme variances (Rousseeuw & Croux, 1993), while the 90th percentile summarises the upper tail without reducing the proposal to one weight. We add one unit to $l^* = \arg\max_l \mathcal{U}^{(l)}$ and draw all newly introduced incoming and outgoing parameters from their prior. The new unit is thus proposed where uncertainty is broad rather than being driven by one outlying weight. High uncertainty determines where to test extra capacity.

Pruning. The "pruning" uses the posterior signal-to-noise ratio $r_w = |\mu_w|/\sigma_w$, following SNR-based weight and structured pruning in variational networks (Blundell et al., 2015; Neklyudov et al., 2017). For unit u in layer l, we average this quantity over incoming and outgoing connections separately, then combine the two directions:

$$r_{u,\text{in}}^{(l)} = \frac{1}{|\mathcal{W}_{u,\text{in}}^{(l)}|}\sum_{w\in\mathcal{W}_{u,\text{in}}^{(l)}}\frac{|\mu_w|}{\sigma_w}, \qquad r_{u,\text{out}}^{(l)} = \frac{1}{|\mathcal{W}_{u,\text{out}}^{(l)}|}\sum_{w\in\mathcal{W}_{u,\text{out}}^{(l)}}\frac{|\mu_w|}{\sigma_w},$$
$$S_u^{(l)} = \sqrt{r_{u,\text{in}}^{(l)} r_{u,\text{out}}^{(l)}}. \tag{3}$$

The geometric mean assigns a small score when either side of the unit carries little posterior signal. A unit is therefore not protected merely because one side has a large average SNR while the other is weak. To compare units across layers, we again standardise within each layer:

$$\widetilde{S}_u^{(l)} = \frac{S_u^{(l)} - m_S^{(l)}}{d_S^{(l)} + \epsilon}, \qquad m_S^{(l)} = \underset{u}{\text{median}}\,S_u^{(l)}, \quad d_S^{(l)} = \underset{u}{\text{median}}\,|S_u^{(l)} - m_S^{(l)}|. \tag{4}$$

Table 1: Methods used for comparison and the mechanism by which each method produces models at different complexity levels.

Method	Description	Complexity variation
Plasticity	Proposed structured phase-free growth and pruning framework	Architecture penalty λ
Static Symmetric	Fixed BNN with equal hidden-layer widths	Initial hidden-layer width
Three-Phase Li et al. (2026)	Structured phase-fixed growing and pruning framework	Initial hidden-layer width
NeST-Inspired Dai et al. (2019)	Unstructured phase-fixed growing and pruning framework	Pruning accuracy floor

We rank all units by $\widetilde{S}_u^{(l)}$ and remove the lowest-scoring fraction ρ, together with their incident connections, to obtain A^-. As with growth, this score defines a proposal rather than an irreversible pruning decision.

3.3 Matched warm-start and selection

After one epoch of variational optimisation of $(A^t, \boldsymbol{\phi}^t)$, we construct A^+ and A^- as above and retain a copy of A^t. Only one of these three candidates will be selected for application to the network at the current step, as described below.

Let $\boldsymbol{\phi}_{A'}$ denote the resulting parameters for candidate $A' \in \{A^t, A^+, A^-\}$. On K shared validation mini-batches, we compute

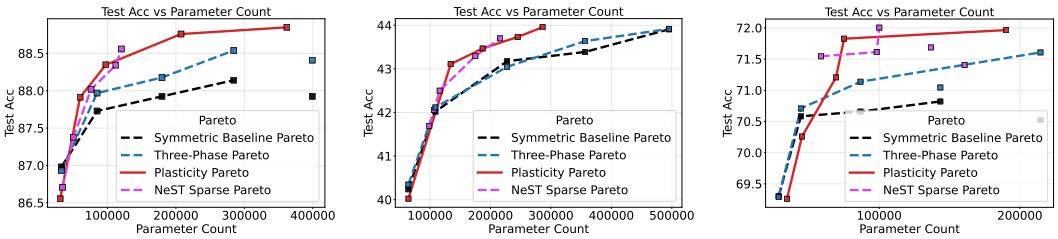
$$s(A') = \frac{1}{K}\sum_{i=1}^{K} \mathcal{J}(A', \boldsymbol{\phi}_{A'}; \mathcal{B}^i_{\text{val}}), \qquad (A^{t+1}, \boldsymbol{\phi}^{t+1}) = \underset{(A', \boldsymbol{\phi}_{A'})}{\arg\max}\, s(A'). \tag{5}$$

Shared training and validation batches reduce avoidable noise in this comparison and make the three scores a paired comparison under the same data realisation. The selected architecture and its warm-started posterior are carried to the next epoch, after which the procedure is repeated (cf. Algorithm 1). No structural action is permanent by design: a model that prunes at one decision can grow later if the validation objective subsequently supports additional capacity.

4 Experimental Setup

We evaluate the proposed plasticity framework on Fashion-MNIST and CIFAR-10. Since the architecture penalty λ influences the training trajectory and the complexity of the architectures selected during training, we evaluate methods through their performance–complexity trade-off rather than at a single model size. Our primary evaluation compares test accuracy against parameter count. As a secondary measure of probabilistic predictive performance, we also report multiclass Brier score against parameter count, with lower values indicating better predictive performance. The corresponding Brier score results are provided in Appendix C. Model complexity is measured by parameter count. All experimental configurations are repeated and aggregated over five independent runs.

We compare the plasticity method against three complementary Bayesian baselines. (i) *Static Symmetric* models use equal widths across hidden layers and are trained with a fixed architecture. To the best of our knowledge, existing work has not considered growing and pruning frameworks in the Bayesian setting, nor phase-free growing and pruning more generally. We therefore adapt two deterministic phase-fixed bidirectional structural adaptation methods as representative baselines for comparison: (ii) the *Three-Phase* baseline, a Bayesian adaptation of a structured phase-fixed growing and pruning method Li et al. (2026) and (iii) the *NeST-Inspired* baseline, a Bayesian adaptation of an unstructured phase-fixed growing and pruning method Dai et al. (2019). Because these methods control model complexity in different ways, Table 1 summarises how each method generates points along the performance–complexity curve.



(a) Fashion-MNIST FNN. (b) CIFAR-10 FNN. (c) CIFAR-10 CNN.

Figure 2: Test accuracy–complexity trade-offs across the three dataset–architecture configurations. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.

Experiments are conducted in both fully-connected and convolutional settings. Both Fashion-MNIST and CIFAR-10 are used to evaluate the fully-connected setting, while only CIFAR-10 is used for the convolutional setting.

Fully Connected Setup. For the fully connected experiments, all methods use the same two-hidden-layer Bayesian specification. Each weight is represented by a mean-field Gaussian variational posterior with a standard normal prior, and one Monte Carlo weight sample is used per forward pass. Each hidden layer is followed by LayerNorm and a SiLU activation.

Convolutional Setup. For the convolutional experiments, all methods use the same two-layer Bayesian CNN specification. The network has two 3×3 convolutions each followed by GroupNorm (one group per channel) and a SiLU activation. The feature map is then globally average-pooled and passed through a Bayesian hidden fully connected layer with SiLU activation and a Bayesian classifier. As in the fully connected setting, every weight is represented by a mean-field Gaussian variational posterior with a standard normal prior, and one Monte Carlo weight sample is drawn per forward pass.

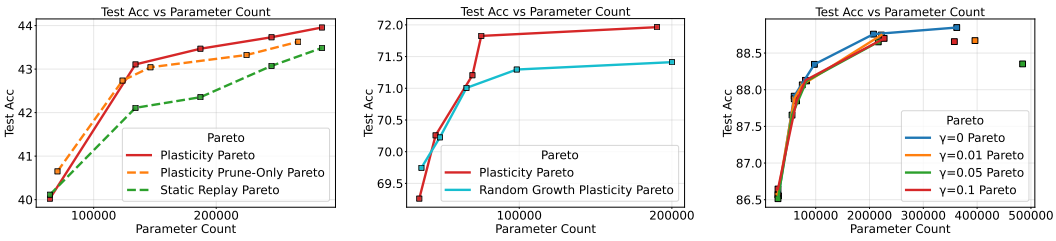
For the plasticity method, both the fully connected and convolutional models are initialised from deliberately overparameterised architectures. This reflects the asymmetry of the structural operators where pruning can remove a fraction of existing structural units at each decision point, whereas growth introduces only a single structural unit.

Exact initial architectures and hyperparameter configurations for each method and dataset setting are provided in Appendix B.

5 Results

Plasticity achieves the best overall predictive performance and the strongest high-capacity performance–complexity trade-off: We compare all methods across the three dataset–architecture configurations. Figure 2 shows the test accuracy–complexity trade-offs for Fashion-MNIST with fully connected networks, CIFAR-10 with fully connected networks, and CIFAR-10 with convolutional networks. Across all three configurations, Plasticity achieves the highest overall mean test accuracy. At comparable larger model sizes, Plasticity also outperforms the Static Symmetric and Three-Phase baselines, demonstrating a stronger performance–complexity trade-off. The Three-Phase baseline consistently underperforms Plasticity, suggesting an advantage of phase-free structural adaptation over a structured phase-fixed alternative. NeST-Inspired remains competitive with Plasticity in terms of test accuracy against parameter count, particularly in the lower-complexity regime. Corresponding Brier score–complexity results are reported in Appendix C.

Structural behaviour of the plasticity framework: Across all three dataset–architecture configurations, structural decisions follow a consistent coarse-to-fine pattern, with pruning dominating early training and retention and occasional growth becoming more frequent as the architecture becomes smaller. The architecture penalty λ also shapes the rate and variability of structural compression. Detailed structural-decision heatmaps and parameter-count trajectories are provided in Appendix D.



(a) Component analysis. (b) Growth criterion. (c) Growth magnitude.

Figure 3: Test accuracy–complexity results for the component and ablation analyses. From left to right: contribution of dynamic adaptation and growth on CIFAR-10 FNN, growth-location criterion on CIFAR-10 CNN, and growth magnitude on Fashion-MNIST FNN. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.

We conduct three complementary ablation and component analyses to examine the contribution of dynamic adaptation and the design choices underlying the growth mechanism.

The gains are not explained by architecture discovery or pruning alone. Using the CIFAR-10 FNN configuration, we compare Plasticity against *Static Replay* and *Prune-Only*. Static Replay retrains from scratch the final architectures discovered by Plasticity while keeping their structure fixed, thereby isolating the benefit of the discovered architecture from that of the dynamic adaptation process. Prune-Only follows the same structural adaptation procedure as Plasticity but disables the growth candidate, allowing us to isolate the contribution of growth. Figure 3(a) shows that Static Replay does not recover the performance of the dynamically trained Plasticity models, suggesting that the gains are not attributable to the discovered architecture alone. Prune-Only also improves over static training but remains slightly inferior to the full Plasticity, indicating an additional contribution from growth.

Uncertainty-based growth outperforms random growth. Using the CIFAR-10 CNN configuration, we compare our proposed plasticity framework against a variant in which the growth location at each structural decision point is selected randomly rather than according to posterior uncertainty. All other components of the framework are kept unchanged. Figure 3(b) shows that uncertainty-based growth achieves a stronger test accuracy–complexity trade-off than random growth, indicating that where additional capacity is introduced is important to the effectiveness of the plasticity mechanism.

Single-unit growth provides the most favourable trade-off. Using the Fashion-MNIST FNN configuration, we compare the proposed single-unit growth rule against proportional growth rates $\gamma \in 0.01, 0.05, 0.1$, where γ denotes the fraction of structural units added to the selected layer at each accepted growth event. The uncertainty-based growth-location criterion and all other components of the framework are kept unchanged. Figure 3(c) shows that performance is broadly similar across growth rates, but single-unit growth achieves the strongest overall test accuracy–complexity trade-off, suggesting that more aggressive growth provides limited additional benefit.

Corresponding Brier score–complexity results for all three analyses are provided in Appendix C.

We additionally evaluate the Fashion-MNIST FNN setting under an **easy-to-hard prior distribution shift**. Training is divided into two phases: Phase 1 contains approximately 90% of examples from the easier class group (Trouser, Sandal, Sneaker, Bag, and Ankle boot) and 10% from the harder group (T-shirt, Pullover, Dress, Coat, and Shirt), with these proportions reversed in Phase 2. Following Variational Continual Learning (VCL) Nguyen et al. (2017), the validation-selected posterior from Phase 1 is used as the prior for Phase 2. After Phase 2, Figure 4 evaluates the final model on the full balanced Fashion-MNIST test set. All other settings remain unchanged from the standard Fashion-MNIST FNN configuration reported in Appendix B.

We compare Plasticity with Static Symmetric and Static Replay, as the phase-fixed Three-Phase and NeST-Inspired baselines do not transfer directly to this two-phase setting. Plasticity achieves a stronger test accuracy–complexity trade-off than both static baselines. Static Replay also improves upon Static Symmetric, suggesting that the architectures discovered by plasticity are benefi-

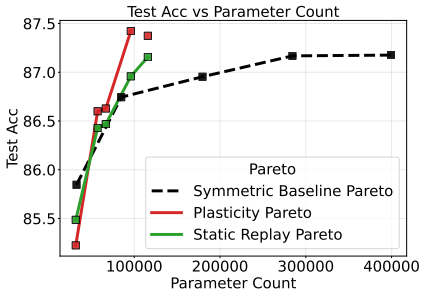


Figure 4: Test accuracy–complexity trade-off under the easy-to-hard Fashion-MNIST distribution shift. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.

cial. However, Plasticity remains superior, indicating that dynamic structural adaptation throughout training provides an additional advantage under distribution shift. The corresponding Brier score–complexity result is reported in Appendix C. Appendix E additionally evaluates retention on a test distribution matching the Phase 1 class proportions.

6 Conclusion

This work presented a phase-free framework for bidirectional structural adaptation in Bayesian neural networks, in which growth, pruning, and retention are considered throughout training under a unified criterion. Across fully connected and convolutional architectures, the method achieved favourable performance–complexity trade-offs compared with static and phase-fixed dynamic baselines, and remained effective under an easy-to-hard distribution shift. Ablation and component analyses further suggest that both dynamic adaptation and the growth mechanism contribute to these gains. Future work could extend the phase-free adaptation principle beyond the Bayesian setting by replacing posterior-based proposal statistics with alternative signals suitable for deterministic neural networks.

Reproducibility Statement

We report the complete training procedure, Bayesian model specifications, structural adaptation settings, baseline configurations, and evaluation protocol used in all experiments. Exact hyperparameter configurations for each dataset–architecture setting are provided in the appendix, together with additional analyses and results. An anonymised implementation of the proposed method and baselines is provided with the submission to facilitate reproduction of the reported experiments.

AI Use Statement

In this work, we used generative AI tools to assist with aspects of implementation, including editing and reviewing software code and providing feedback on experimental setup and methodology. We did not use generative AI tools to generate synthetic datasets or to formulate or prove mathematical claims. All implementation-related AI outputs were manually reviewed and tested before use.

Generative AI tools were also used as an advanced language editing aid to improve grammatical correctness, clarity, and readability without changing the underlying technical content. All AI-assisted edits were reviewed and revised by the authors.

Finally, generative AI tools were used for literature discovery. References were checked against the original sources. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL http://jmlr.org/papers/v13/bergstra12a.html.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016.

Xiaoliang Dai, Hongxu Yin, and Niraj K Jha. Nest: A neural network synthesis tool based on a grow-and-prune paradigm. *IEEE Transactions on Computers*, 68(10):1487–1497, 2019.

Charles De Lambilly and Stefan Duffner. Growing networks with autonomous pruning. *arXiv preprint arXiv:2603.19759*, 2026.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: a survey. *J. Mach. Learn. Res.*, 20(1):1997–2017, January 2019. ISSN 1532-4435.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2943–2952, 2020.

Utku Evci, Bart van Merriënboer, Thomas Unterthiner, Fabian Pedregosa, and Max Vladymyrov. Gradmax: Growing neural networks using gradient information. In *International Conference on Learning Representations*, 2022.

Zoubin Ghahramani and Thomas L. Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in Neural Information Processing Systems*, volume 18, 2005.

Magdalena Götz and Wieland B Huttner. The cell biology of neurogenesis. *Nature reviews Molecular cell biology*, 6(10):777–788, 2005.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Shaoyi Huang, Haowen Fang, Kaleel Mahmood, Bowen Lei, Nuo Xu, Bin Lei, Yue Sun, Dongkuan Xu, Wujie Wen, and Caiwen Ding. Neurogenesis dynamics-inspired spiking neural network training acceleration. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6. IEEE, 2023.

Siddhant M. Jayakumar, Razvan Pascanu, Jack W. Rae, Simon Osindero, and Erich Elsen. Top-kast: Top-k always sparse training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2575–2583, 2015.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations*, 2017.

Yupei Li, Manuel Milling, and Björn W Schuller. Neuroplasticity in artificial intelligence–an overview and inspirations on drop in & out learning. *arXiv preprint arXiv:2503.21419*, 2025.

Yupei Li, Shuaijie Shao, Manuel Milling, and Björn Schuller. Enhancing efficiency and performance in deepfake audio detection through neuron-level dropin & neuroplasticity mechanisms. *arXiv preprint arXiv:2603.24343*, 2026.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. In *Advances in Neural Information Processing Systems*, volume 34, pp. 9908–9922, 2021.

Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. *Advances in neural information processing systems*, 30, 2017.

David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992. doi: 10.1162/neco.1992.4.3.448.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phu Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1):2383, 2018.

Pavlo Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4646–4655, 2019.

Eric T. Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *International Conference on Learning Representations*, 2017.

Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Structured bayesian pruning via log-normal multiplicative noise. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Martin C Raff, Barbara A Barres, Julia F Burne, Harriet S Coles, Yasuki Ishizaki, and Michael D Jacobson. Programmed cell death and the control of cell survival: lessons from the nervous system. *Science*, 262(5134):695–700, 1993.

Peter J Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283, 1993.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Anirudh Sowrirajan, Pranav Srinivasan, Sundari Avanthikaa Srinivasan, and S Prasanna Devi. Enhancing neurofuzzy plasticity: A fusion of lstm and artificial neurogenesis. In *2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI)*, pp. 150–154. IEEE, 2024.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/41bfd20a38bb1b0bec75acf0845530a7-Paper.pdf.

Lemeng Wu et al. Firefly neural architecture descent: A general approach for growing neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22373–22383, 2020.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Zixing Zhang, Jing Han, Kun Qian, and Björn Schuller. Evolving Learning for Analysing Mood-Related Infant Vocalisation. In *Interspeech 2018*, pp. 142–146, 2018. doi: 10.21437/Interspeech.2018-1914.

A Derivation of the Complexity-Aware Variational Objective

We derive the penalised ELBO used for structural selection by extending the standard variational treatment of Bayesian neural networks to include the architecture A as a model variable. Let

$$q(\boldsymbol{\theta}, A) = q_{\boldsymbol{\psi}}(A)\, q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A)$$

denote the joint variational approximation over weights $\boldsymbol{\theta}$ and architecture A.

Starting from the KL divergence between the variational distribution and the true joint posterior,

$$\begin{aligned} &D_{\mathrm{KL}}\left(q(\boldsymbol{\theta}, A) \| p(\boldsymbol{\theta}, A \mid \mathcal{D})\right) \\ &\quad = \mathbb{E}_{q(\boldsymbol{\theta}, A)}\left[\log q(\boldsymbol{\theta}, A) - \log p(\boldsymbol{\theta}, A \mid \mathcal{D})\right]. \end{aligned} \tag{6}$$

Applying Bayes' rule gives

$$\begin{aligned} &= \mathbb{E}_{q(\boldsymbol{\theta}, A)}\Big[\log q(\boldsymbol{\theta}, A) - \log p(\mathcal{D} \mid \boldsymbol{\theta}, A) \\ &\qquad - \log p(\boldsymbol{\theta}, A)\Big] + \log p(\mathcal{D}). \end{aligned} \tag{7}$$

Using

$$q(\boldsymbol{\theta}, A) = q_{\boldsymbol{\psi}}(A) q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A)$$

and

$$p(\boldsymbol{\theta}, A) = p(\boldsymbol{\theta} \mid A) p(A),$$

the terms can be regrouped as

$$\begin{aligned} &D_{\mathrm{KL}}\left(q(\boldsymbol{\theta}, A) \| p(\boldsymbol{\theta}, A \mid \mathcal{D})\right) \\ &= \mathbb{E}_{q_{\boldsymbol{\psi}}(A)}\Bigg[D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \| p(\boldsymbol{\theta} \mid A)\right) \\ &\qquad - \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A)}\left[\log p(\mathcal{D} \mid \boldsymbol{\theta}, A)\right]\Bigg] \\ &\quad + D_{\mathrm{KL}}\left(q_{\boldsymbol{\psi}}(A) \| p(A)\right) + \log p(\mathcal{D}). \end{aligned} \tag{8}$$

Since the KL divergence on the left-hand side is non-negative,

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q_{\boldsymbol{\psi}}(A)} \Bigg[\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|A)} \left[\log p(\mathcal{D}_y \mid \mathcal{D}_x, \boldsymbol{\theta}, A) \right] - D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \| p(\boldsymbol{\theta} \mid A) \right) \Bigg] - D_{\mathrm{KL}} \left(q_{\boldsymbol{\psi}}(A) \| p(A) \right). \tag{9}$$

This yields the joint ELBO

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{q_{\boldsymbol{\psi}}(A)} \Bigg[\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|A)} \left[\log p(\mathcal{D}_y \mid \mathcal{D}_x, \boldsymbol{\theta}, A) \right] - D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \| p(\boldsymbol{\theta} \mid A) \right) \Bigg] - D_{\mathrm{KL}} \left(q_{\boldsymbol{\psi}}(A) \| p(A) \right). \tag{10}$$

A.1 Point-Mass Approximation over Architectures

Exact variational inference over architectures is combinatorial. In the proposed framework, structural decisions are instead made over a finite set of candidate architectures at each decision point. We therefore approximate the variational distribution over architectures by a point mass,

$$q_{\boldsymbol{\psi}}(A) \approx \delta_{A'}, \tag{11}$$

where A' denotes the candidate architecture being evaluated.

Under this approximation, the expectation over architectures collapses to the candidate A', and the architecture-level KL becomes

$$\begin{aligned} D_{\mathrm{KL}} \left(\delta_{A'} \| p(A) \right) &= \mathbb{E}_{\delta_{A'}} \left[\log \delta_{A'}(A) - \log p(A) \right] \\ &= -\log p(A') + C, \end{aligned} \tag{12}$$

where C is independent of the candidate architecture and can therefore be ignored when comparing candidates.

A.2 Architecture Prior

To encode a preference for parsimonious architectures, we define

$$p(A) \propto \exp[-\lambda N(A)], \tag{13}$$

where $N(A)$ denotes the number of parameters in architecture A and $\lambda \geq 0$ controls the strength of the complexity preference. Consequently,

$$-\log p(A) = \lambda N(A) + C', \tag{14}$$

where C' is independent of A.

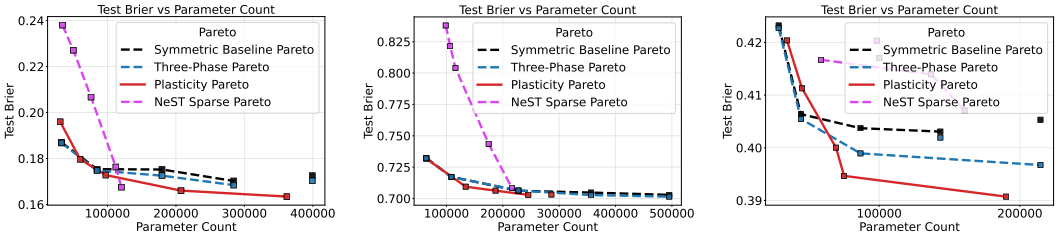
Substituting this result into the joint ELBO and discarding additive constants gives

$$\begin{aligned} \mathcal{L}(A, \boldsymbol{\phi}) =& \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|A)} \left[\log p(\mathcal{D}_y \mid \mathcal{D}_x, \boldsymbol{\theta}, A) \right] \\ & - D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \| p(\boldsymbol{\theta} \mid A) \right) - \lambda N(A). \end{aligned} \tag{15}$$

In practice, following the training objective used for the Bayesian weight posterior, we introduce a coefficient β on the weight-level KL term. The final objective used for both weight optimisation and structural candidate selection is therefore

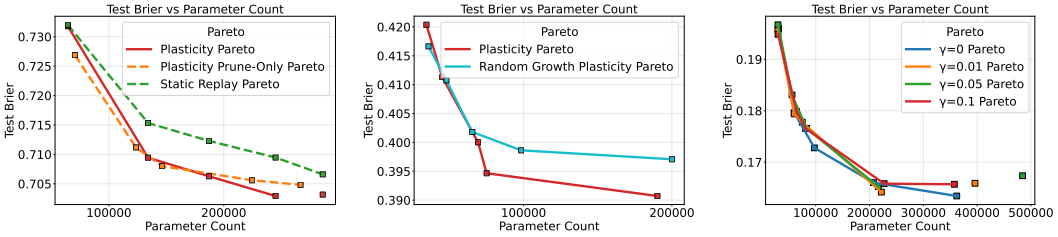
$$\begin{aligned} \mathcal{L}_{\mathrm{pen}}(A, \boldsymbol{\phi}) =& \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\theta}|A)} \left[\log p(\mathcal{D}_y \mid \mathcal{D}_x, \boldsymbol{\theta}, A) \right] \\ & - \beta D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta} \mid A) \| p(\boldsymbol{\theta} \mid A) \right) - \lambda N(A). \end{aligned} \tag{16}$$

The first two terms form the variational learning objective over network weights, while the final term penalises architectural complexity. The hyperparameter λ therefore controls the pressure toward smaller architectures: larger values favour more parsimonious models, whereas smaller values permit additional capacity when it improves predictive performance.



(a) Fashion-MNIST FNN. (b) CIFAR-10 FNN. (c) CIFAR-10 CNN.

Figure 5: Brier score–complexity trade-offs for the three main dataset–architecture configurations. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.



(a) Component analysis. (b) Growth-location criterion. (c) Growth magnitude.

Figure 6: Brier score–complexity results for the component and ablation analyses. From left to right: contribution of dynamic adaptation and growth on CIFAR-10 FNN, growth-location criterion on CIFAR-10 CNN, and growth magnitude on Fashion-MNIST FNN. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.

B Configuration of the Hyperparameters

Table 2 summarises the hyperparameter configurations used across all dataset–architecture settings and comparison methods. In addition to the fixed training and structural adaptation settings, the table reports the hyperparameter sweeps used in the main experiments to generate different points along the performance–complexity curves. Values shown in braces indicate the set of configurations evaluated for each sweep.

C Additional Brier Score Results

C.1 Main Performance Comparisons

Figure 5 reports the Brier score–complexity trade-offs corresponding to the main accuracy results in Section 5. The overall comparison is consistent with the accuracy results.

C.2 Ablation and Component Analysis

Figure 6 reports the Brier score–complexity results corresponding to the component and ablation analyses in the main text. The overall comparison is consistent with the accuracy results.

C.3 Distribution Shift

Figure 7 reports the Brier score–complexity trade-off for the easy-to-hard Fashion-MNIST distribution-shift experiment. The overall comparison is consistent with the accuracy results.

Table 2: Hyperparameter configurations for all experimental settings. Values shown in braces indicate hyperparameter sweeps used to generate different points along the performance–complexity curve.

Setting	Fashion-MNIST FNN	CIFAR-10 FNN	CIFAR-10 CNN
Common training settings			
Optimiser	Adam	Adam	Adam
Learning rate	5×10^{-3}	1×10^{-3}	1×10^{-2}
Batch size	256	128	64
KL weight β	0.002	0.002	0.01
Plasticity			
Initial architecture	$[500, 500]$	$[500, 500]$	$[300f, 300f]$
Architecture penalty λ	$\{0, 10^{-7}, 5 \times 10^{-7}, 10^{-6}, 5 \times 10^{-6}\}$	$\{0, 10^{-7}, 5 \times 10^{-7}, 10^{-6}, 5 \times 10^{-6}\}$	$\{10^{-7}, 5 \times 10^{-7}, 10^{-6}, 2.5 \times 10^{-6}, 5 \times 10^{-6}\}$
Structural decision frequency	Every epoch	Every epoch	Every epoch
Warm-start steps M	64	64	32
Isolated growth steps M_1	16	32	16
Pruning fraction ρ	0.1	0.1	0.1
Validation batches N	20	20	20
Warm-start learning-rate multiplier	0.4	0.4	0.4
Static Symmetric			
Architecture sweep	$\{[20, 20], [50, 50]$ $[100, 100], [150, 150]$ $[200, 200]\}$	$\{[30, 30], [50, 50]$ $[100, 100], [150, 150]$ $[200, 200]\}$	$\{[30f, 30f], [40f, 40f]$ $[60f, 60f], [80f, 80f]$ $[100f, 100f]\}$
Three-Phase			
Architecture Sweep	$\{[20, 20], [50, 50]$ $[100, 100], [150, 150]$ $[200, 200]\}$	$\{[30, 30], [50, 50]$ $[100, 100], [150, 150]$ $[200, 200]\}$	$\{[30f, 30f], [40f, 40f]$ $[60f, 60f], [80f, 80f]$ $[100f, 100f]\}$
Growth Layer	Hidden layer 2	Hidden layer 2	Hidden layer 2
NeST-Inspired			
Initial architecture	$[300, 100]$	$[300, 100]$	$[300f, 100f]$
Initial Sparsity	0.1	0.1	0.1
Reference Validation Accuracy	89.0	44.0	71.5
Growth interval	2 epochs	2 epochs	2 epochs
Connection growth fraction	1%	1%	1%
Growth Ratio	0.4	0.4	-
Birth Strength	0.4	0.4	-
Number of feature map candidates	-	-	12
Outgoing activation fraction for new feature map	-	-	0.3
Pruning accuracy floor	$\{89.0, 88.5, 88.0, 87.5, 87.0\}$	$\{44.0, 43.5, 43.0, 42.5, 42.0\}$	$\{71.5, 71.25, 71.0, 70.75, 70.5\}$
Pruning interval	2 epochs	2 epochs	2 epochs
Pruning fraction	1%	1%	1%

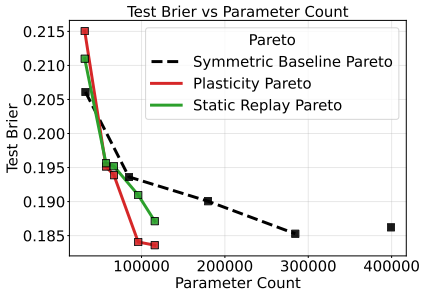


Figure 7: Brier score versus parameter count under the easy-to-hard Fashion-MNIST distribution shift. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration.

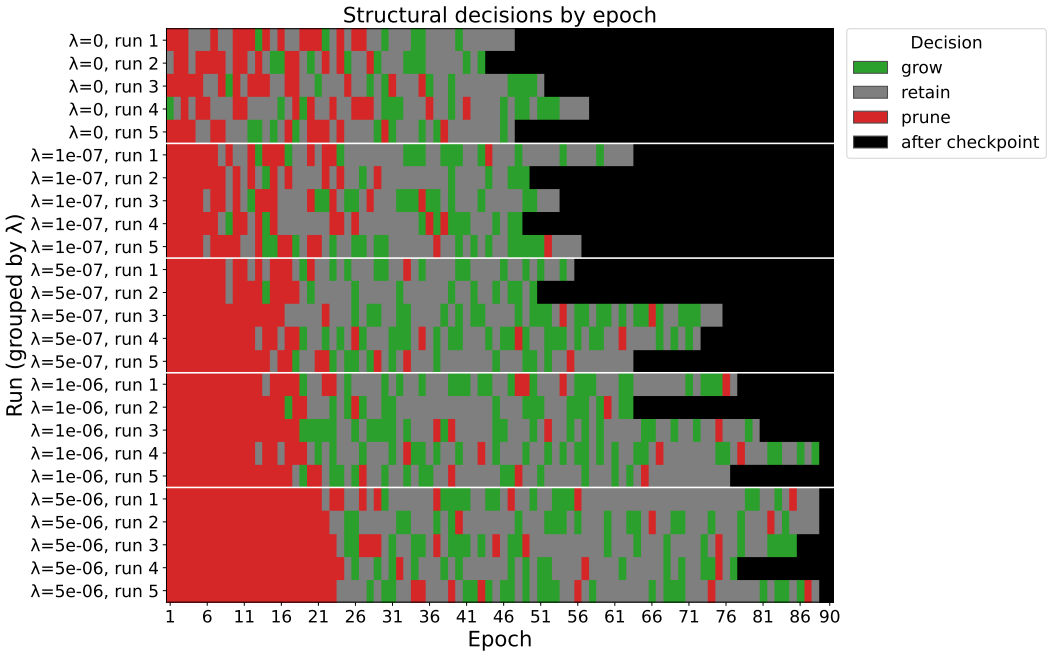


Figure 8: Structural decisions across training for the Fashion-MNIST FNN configuration under different values of λ and five independent runs. Each cell indicates whether growth, pruning, or retention is selected at a structural decision point, while black cells denote epochs occurring after the validation-selected checkpoint.

D Additional Structural Analysis

D.1 Structural Decision Patterns

Figures 8, 9, and 10 show the structural decisions selected throughout training for the Fashion-MNIST FNN, CIFAR-10 FNN, and CIFAR-10 CNN settings, respectively. Across all three configurations, pruning dominates the early stages of training, when the models remain highly overparameterised, whereas retention and occasional growth become more frequent as the architectures become smaller. This suggests a consistent coarse-to-fine adaptation process in which pruning first removes excess capacity, after which retention and single-unit growth provide finer structural adjustments.

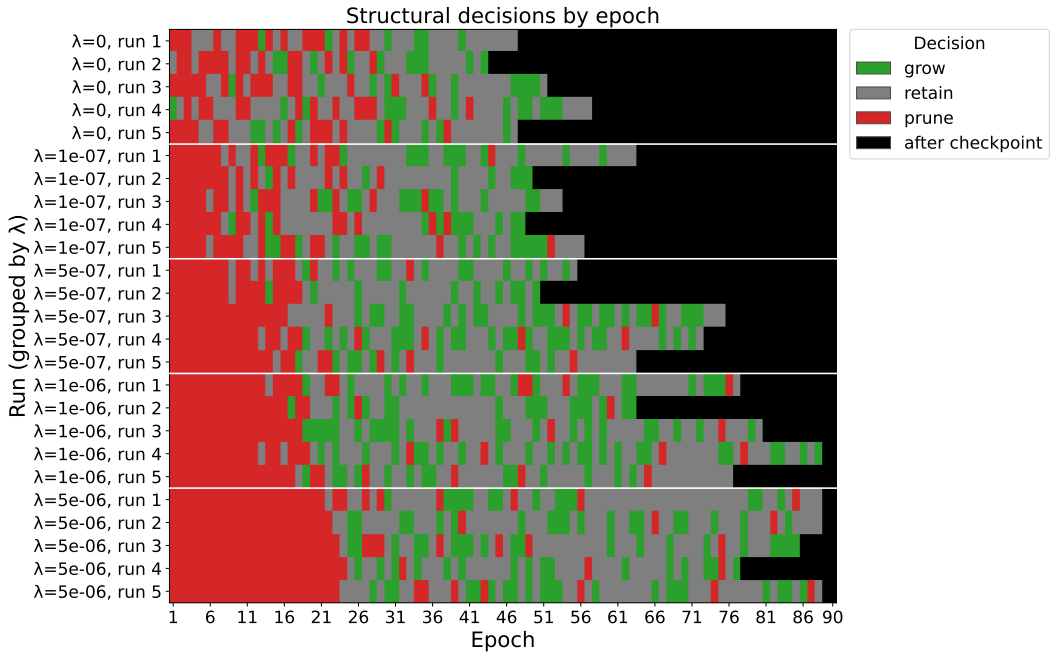


Figure 9: Structural decisions across training for the CIFAR-10 FNN configuration under different values of λ and five independent runs. Each cell indicates whether growth, pruning, or retention is selected at a structural decision point, while black cells denote epochs occurring after the validation-selected checkpoint.

D.2 Effect of the Architecture Penalty

Figures 11, 12, and 13 show the evolution of parameter count throughout training under different values of λ. Across all three settings, smaller values of λ permit greater variability in structural trajectories, whereas larger values encourage faster and more consistent compression.

Substantial compression also occurs at $\lambda = 0$, despite the absence of an explicit architecture-complexity penalty. One possible contributor is the weight-level KL term, which remains active and penalises posterior deviation from the prior across all Bayesian parameters. Overall, these results indicate that λ shapes both the rate and consistency of structural adaptation rather than simply determining a fixed final model size.

E Forgetting under Prior Distribution Shift

To complement the balanced-test evaluation reported in Figure 4, we additionally evaluate the final model obtained after Phase 2 on a test distribution matching the class proportions of Phase 1. Specifically, the test set contains approximately 90% examples from the easier class group and 10% from the harder group. This provides an additional measure of how well each method retains performance on the earlier distribution after adapting to the Phase 2 shift. Figure 14 shows the corresponding performance–complexity trade-offs for test accuracy and Brier score. For both test accuracy and Brier score against parameter count, Plasticity achieves the strongest performance–complexity trade-off when the final Phase 2 model is evaluated on a test distribution matching the Phase 1 proportions. It consistently outperforms both Static Symmetric and Static Replay, suggesting that dynamic structural adaptation may help preserve representations learnt during Phase 1 even after the model adapts to the shifted Phase 2 distribution.

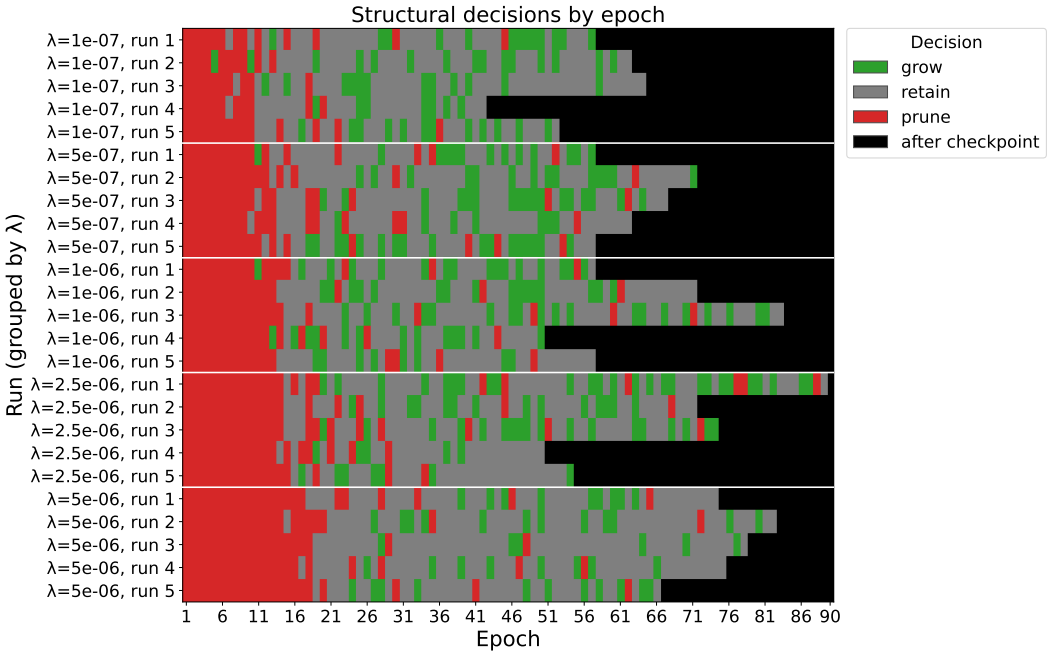


Figure 10: Structural decisions across training for the CIFAR-10 CNN configuration under different values of λ and five independent runs. Each cell indicates whether growth, pruning, or retention is selected at a structural decision point, while black cells denote epochs occurring after the validation-selected checkpoint.

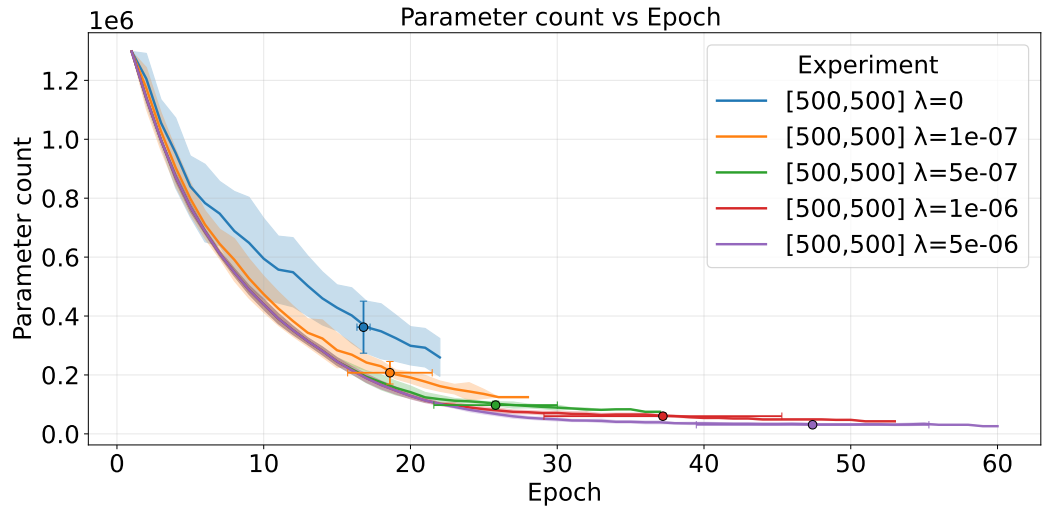


Figure 11: Evolution of parameter count throughout training for the Fashion-MNIST FNN configuration under different values of λ, aggregated over five independent runs. Curves show the mean across five runs and shaded regions indicate$\pm$ one standard deviation. Markers denote the validation-selected checkpoints.

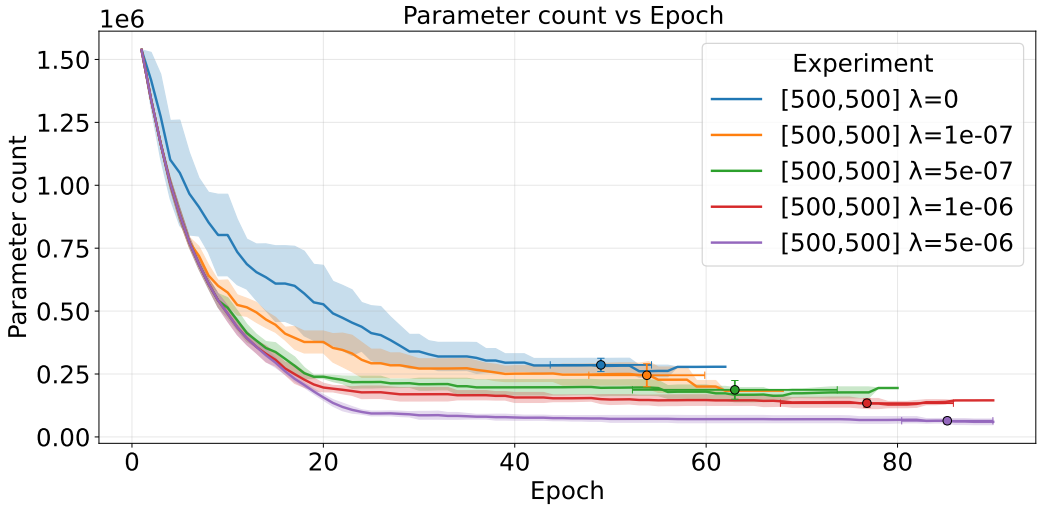


Figure 12: Evolution of parameter count throughout training for the CIFAR-10 FNN configuration under different values of λ, aggregated over five independent runs. Curves show the mean across five runs and shaded regions indicate$\pm$ one standard deviation. Markers denote the validation-selected checkpoints.

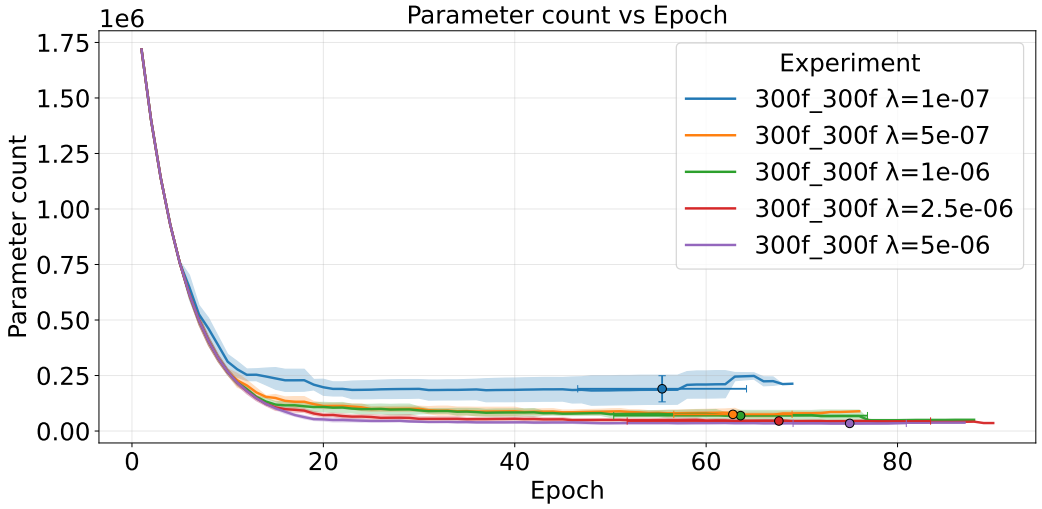
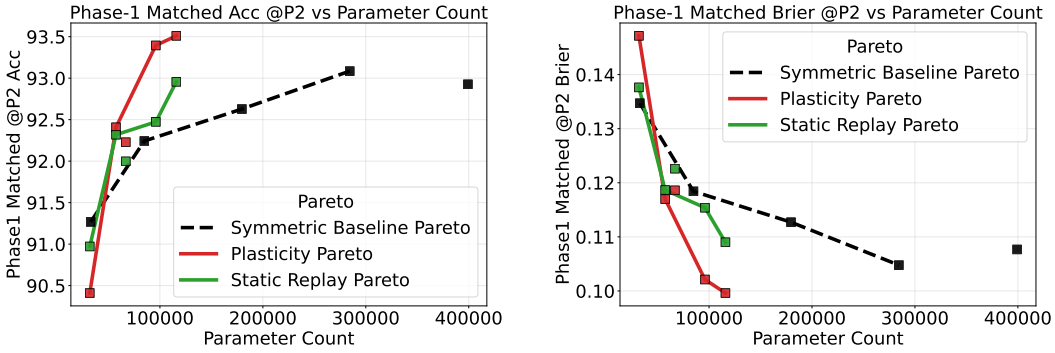


Figure 13: Evolution of parameter count throughout training for the CIFAR-10 CNN configuration under different values of λ, aggregated over five independent runs. Curves show the mean across five runs and shaded regions indicate$\pm$ one standard deviation. Markers denote the validation-selected checkpoints.



(a) Test accuracy versus parameter count.

(b) Test Brier score versus parameter count.

Figure 14: Performance–complexity trade-off after Phase 2 when the final model is evaluated on a test distribution matching the Phase 1 class proportions. Results are aggregated over five independent runs. Curves show the mean trajectory across runs, while markers denote the mean result at each evaluated configuration. This evaluation provides an additional view of retention on the earlier distribution after adaptation to the easy-to-hard shift.