NEURAL BAYESIAN SEQUENTIAL ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Neural Bayesian Sequential Routing (NBSR), a framework that turns neural inference into a sequential Bayesian decision process. NBSR combines four key ideas: a persistent global knowledge oracle, a hierarchical graph of routers and experts, an explicit Dirichlet belief state updated by accumulated neural evidence, and hard belief-conditioned routing trained end-to-end with Gumbel-Softmax. Together, these components unify conditional computation, uncertainty estimation, and sequential decision-making in one architecture. The evolving belief state enables early exit, OOD abstention, cost-aware evidence acquisition, and transparent path-wise audit trails. We prove monotone growth of Dirichlet precision and bounded predictive variance under positive evidence, formalizing sequential hypothesis sharpening, and show Bayes-optimal recovery under idealized assumptions. Extensions with recurrent memory and autoregressive routing support partially observable control and Bayesian experimental design. Across vision, diagnosis, language modeling, control, and active triage, NBSR achieves competitive predictive performance while providing uncertainty-aware, interpretable, and resource-rational inference.

1 INTRODUCTION

Human decision-making is sequential, distributional, and hierarchical (Kahneman, 2011; Wald, 1945): we accumulate evidence step by step (Bonawitz et al., 2010), condition each decision on what we have already concluded, and let broad hypotheses sharpen into confident conclusions. Standard deep networks instead process all features at once in a single dense pass (LeCun et al., 2015; He et al., 2016). This offers little visibility into how evidence is acquired or when computation could stop (Lipton, 2018; Rudin, 2019), and it spends the same computation on easy and hard inputs (Graves, 2017; Han et al., 2022). Mixture-of-Experts (MoE) models introduce conditional computation (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022), but routing is typically a single-step, input-conditioned soft weighting: it keeps no belief state and cannot sequentially accumulate evidence or re-evaluate uncertainty.

We introduce **Neural Bayesian Sequential Routing (NBSR)**, which models inference as an evidence-accumulating traversal of a hierarchical graph of routers and experts. An input is embedded once into a persistent *global knowledge oracle*; at each step, a router conditioned on the oracle and on the current belief selects one expert, which queries the oracle and emits a strictly positive evidence vector. The evidence is added to a Dirichlet belief state by conjugate addition (Gelman et al., 2013; Bishop, 2006), so the belief sharpens along the path. Hard, path-dependent routing is trained end-to-end with the Gumbel-Softmax straight-through estimator (Jang et al., 2017; Maddison et al., 2017; Bengio et al., 2013). Because the belief state is explicit, its entropy and precision directly support early exiting, out-of-distribution (OOD) abstention, and cost-aware evidence acquisition, and every prediction comes with an additive, per-expert audit trail. Our contributions are:

- **A Bayesian sequential routing architecture** that couples a persistent oracle, a hierarchical router–expert graph, and a Dirichlet belief state updated by exact conjugate addition, trained end-to-end with hard, belief-conditioned routing (Section 3).
- **Theoretical guarantees**: strict monotone growth of Dirichlet precision with an $\mathcal{O}(1/\alpha_0)$ variance bound under positive evidence, epistemic collapse for in-distribution inputs,

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.

Bayes-optimal recovery under idealized assumptions, and a topological bias–variance analysis (Section 4).

- **Extensions**: NBSR-Mem, with recurrent memory for partially observable control, and AR-NBSR, an autoregressive state machine that performs cost-aware sequential Bayesian experimental design without combinatorial explosion (Sections 5.4 and 5.5).
- **Empirical evaluation** across vision (CIFAR-10), structured medical diagnosis, language modeling, POMDP control, and active clinical triage, showing competitive accuracy together with calibrated uncertainty, early exit, OOD abstention, and interpretable routing traces (Section 5).

2 Related Work

Uncertainty in deep networks. Bayesian neural networks place distributions over weights, approximated by variational inference (Blundell et al., 2015) or Monte Carlo dropout (Gal & Ghahramani, 2016), and capture parameter uncertainty rather than a sequentially accumulated belief. Evidential deep learning (EDL) (Sensoy et al., 2018) predicts Dirichlet concentrations in a single forward pass, grounded in subjective logic (Jøsang, 2016). NBSR distributes evidence extraction across a dynamically routed hierarchy, turning evidential inference into a sequential process, and uses exact conjugacy (Raiffa & Schlaifer, 1961; Gelman et al., 2013) rather than sampling.

Conditional computation and trees. CART partitions raw inputs with hard splits (Breiman et al., 1984); hierarchical mixtures of experts (HME) use soft gates at internal nodes and predict only at the leaves (Jordan & Jacobs, 1993; Waterhouse & Robinson, 1995); sparse MoE layers route tokens with top-k gating and load-balancing losses (Shazeer et al., 2017; Fedus et al., 2022); and adaptive computation and dynamic networks vary depth per input (Graves, 2017; Han et al., 2022). In NBSR every visited node, internal or terminal, is an evidence extractor; routing is hard and conditioned on the belief; experts query a shared semantic oracle rather than raw features; and evidence is *added* rather than probabilities multiplied, which permits explicit epistemic uncertainty (a detailed comparison is in Section C).

Sequential decisions and experimental design. Early exit and abstention relate to sequential hypothesis testing (Wald, 1945) and to OOD detection (Hendrycks & Gimpel, 2017). Planning under partial observability is formalized by POMDPs (Kaelbling et al., 1998), and Bayesian optimal experimental design (BOED) selects measurements that maximize expected information gain (Sebastiani & Wynn, 2000; Foster et al., 2019). NBSR's conjugate belief state makes the realized information gain of each routing step available in closed form.

3 Neural Bayesian Sequential Routing

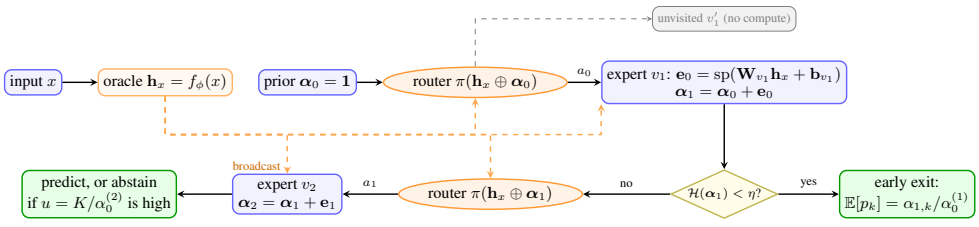


Figure 1: NBSR inference on a depth-2 graph. The oracle $\mathbf{h}_x$ is computed once and queried by every visited router and expert. Each router selects one child using $\mathbf{h}_x$ and the current belief $\boldsymbol{\alpha}_t$ (hard Gumbel-Softmax during training, $\arg\max$ at test time); the chosen expert adds positive evidence (sp: Softplus) to the Dirichlet belief. Inference stops early when the entropy falls below η, or abstains when epistemic uncertainty stays high. The full training graph with gradient paths is in Fig. 8, Section A.

Decision graph and oracle. The model is a directed acyclic super-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of maximal depth L and branching factor $\bar{W}$, with a root v_0 that acts only as a router. An input is embedded once by a shared backbone, $\mathbf{h}_x = f_\phi(x) \in \mathbb{R}^d$, and this *global knowledge oracle* is broadcast to every

node, so each decision has access to the full context of the input, unlike layer-by-layer processing (Fig. 1). The graph topology is decoupled from the K-dimensional outcome space: every expert emits a full K-dimensional evidence vector but specializes in a sub-population.

Belief state. The belief over outcomes is a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha}_t)$, initialized to the uniform prior $\boldsymbol{\alpha}_0 = \mathbf{1}$. With precision $\alpha_0^{(t)} = \sum_k \alpha_{t,k}$, the expected class probability is $\mathbb{E}[p_k] = \alpha_{t,k}/\alpha_0^{(t)}$, the subjective-logic epistemic uncertainty is $u = K/\alpha_0^{(t)}$ (Jøsang, 2016; Sensoy et al., 2018), and $\mathcal{H}(\boldsymbol{\alpha}_t)$ is the Dirichlet differential entropy.

Belief-conditioned hard routing. At node v_t, a router produces logits over the $|\mathcal{A}(v_t)|$ children from the oracle, the belief, and optional side information,

$$\mathbf{z}_t = \mathrm{MLP}_{\theta_{v_t}}\big([\mathbf{h}_x \oplus \boldsymbol{\alpha}_t \oplus \boldsymbol{\varepsilon}_t]\big), \tag{1}$$

so the path depends on what the model has already concluded. During training, we sample Gumbel-perturbed, temperature-τ relaxations $\boldsymbol{\pi}$ of $\mathbf{z}_t$ and use the straight-through estimator: the forward pass executes only the $\arg\max$ child, while gradients flow through $\boldsymbol{\pi}$ (Jang et al., 2017; Maddison et al., 2017; Bengio et al., 2013). At test time, routing is deterministic ($\arg\max \mathbf{z}_t$).

Evidence extraction and conjugate update. The selected expert v_{t+1} queries the oracle with its own filter and emits strictly positive evidence, which is added to the belief:

$$\mathbf{e}_t = \sigma_+\big(\mathbf{W}_{v_{t+1}}\mathbf{h}_x + \mathbf{b}_{v_{t+1}}\big) \in \mathbb{R}^K_{>0}, \qquad \boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t. \tag{2}$$

Treating $\mathbf{e}_t$ as pseudo-counts, this is the continuous analogue of the Dirichlet–Categorical conjugate update. For leaves, σ_+ is the unbounded Softplus; intermediate nodes may use a scaled sigmoid, which caps the evidence they can contribute (an *epistemic confidence budget*), so that early, broad hypotheses remain appropriately hesitant.

Training objective. With terminal step T on the sampled trajectory $\mathcal{T}$, we minimize the NLL of the expected terminal prediction plus a trajectory entropy penalty that rewards early sharpening,

$$\mathcal{L}(x, y^*) = -\log \frac{\alpha_{T,y^*}}{\sum_k \alpha_{T,k}} + \lambda \sum_{v_t \in \mathcal{T}} \mathcal{H}(\boldsymbol{\alpha}_t), \tag{3}$$

averaged over the data, and jointly over backbone, routers, and experts. Only routers and experts on the active path receive gradients, so experts specialize while routers learn from the relaxed counterfactual probabilities (Algorithm 1). Hard routing assigns each expert full responsibility for its evidence, avoiding the blended, weakly specialized experts of soft routing (Section A).

Inference: early exit and abstention. Traversal stops as soon as $\mathcal{H}(\boldsymbol{\alpha}_t) < \eta$, and the prediction is $\mathbb{E}[p_k]$ under the current belief, so easy inputs take shallow paths. If $u = K/\alpha_0$ remains above a safety threshold after the full traversal, the model abstains. The expected cost is $\mathcal{O}\big(C_{\text{backbone}} + \mathbb{E}[T(x)]\,(d\,d_h + dK)\big)$: although the super-graph has $\mathcal{O}(\bar{W}^L)$ nodes, only one path of at most L nodes is executed, and activation memory grows linearly with depth.

4 Theoretical Analysis

Theorem 1 (Strict precision monotonicity and bounded variance)**.** *For any input x and any routing trajectory, if every expert's activation σ_+ is strictly positive, then $\alpha_0^{(t+1)} > \alpha_0^{(t)}$ for all t, and the variance of every marginal is bounded by $\mathrm{Var}[p_k^{(t)}] \le \frac{1}{4(\alpha_0^{(t)}+1)}$, which is $\mathcal{O}(1/\alpha_0^{(t)})$.*

Proof. Since $e_{t,k} > 0$ for all k, $\alpha_0^{(t+1)} = \alpha_0^{(t)} + \sum_k e_{t,k} > \alpha_0^{(t)}$. The Dirichlet marginal variance is $\mathrm{Var}[p_k^{(t)}] = \alpha_{t,k}(\alpha_0^{(t)} - \alpha_{t,k})/\big((\alpha_0^{(t)})^2(\alpha_0^{(t)}+1)\big)$; by AM–GM the numerator is at most $(\alpha_0^{(t)})^2/4$, which gives the bound. □

Corollary 1 (Epistemic collapse)**.** *Under the conditions of Theorem 1, the epistemic uncertainty $u = K/\alpha_0^{(t)}$ strictly decreases along every trajectory. Inputs that elicit near-zero evidence (e.g., OOD inputs) leave α_0 close to its prior value K, keep u close to 1, and can therefore be detected by thresholding u.*

The differential entropy is *not* guaranteed to decrease at every step, since evidence that contradicts the current belief can move its mass and briefly raise entropy, but it tends to $-\infty$ as $\alpha_0 \to \infty$.

Theorem 2 (Bayes-optimal consistency)**.** *Assume that the backbone and experts have unlimited functional capacity, that $\lambda = 0$, and that the expected objective in Eq.* (3) *attains its global minimum. Then the terminal Dirichlet mean recovers the true conditional distribution: $\mathbb{E}[p_k \mid \boldsymbol{\alpha}_T(x)] = P(y = k \mid x)$ almost everywhere.*

Proof sketch. With $\lambda = 0$, the expected loss is the cross-entropy between $P(y \mid x)$ and $q(y \mid x) = \alpha_{T,y}(x)/\alpha_0^{(T)}(x)$, which equals $\mathcal{H}(P(y \mid x)) + \mathrm{KL}(P \,\|\, q)$; by Gibbs' inequality it is minimized if and only if $q = P$, which is attainable under unlimited capacity (Cybenko, 1989; Hornik, 1991). The full proof is in Section B. □

The roles of λ and η.

Proposition 1 (λ accelerates information acquisition)**.** *Because the penalty $\lambda \sum_{v_t \in \mathcal{T}} \mathcal{H}(\boldsymbol{\alpha}_t)$ sums entropy over every step, it is reduced by lowering entropy* early *in the trajectory; increasing λ therefore incentivizes early experts to extract larger, more discriminative evidence, yielding a steeper entropy descent.*

The test-time threshold η bounds the tolerated uncertainty, since a lower η forces deeper traversal; together, λ (at training time) and η (at test time) control the accuracy–computation trade-off.

Topological bias–variance trade-off. Under the standard decomposition of expected NLL into noise, bias $\mathrm{KL}(P^* \| \bar{P}_{\mathcal{G}})$, and variance $\mathbb{E}_{\mathcal{D}}\mathrm{KL}(\bar{P}_{\mathcal{G}} \| P_{\mathcal{G}}(\cdot; \mathcal{D}))$ (Heskes, 1998), greater width $\bar{W}$ and depth L reduce bias but fragment the data (about $|\mathcal{D}|/\bar{W}^l$ samples per node at depth l) and compound routing stochasticity, which increases variance. Implicit pruning (routers learn to ignore unhelpful experts) and entropy-based early exit make the effective width and depth sample-dependent: easy inputs use shallow, well-populated paths and hard inputs use deeper, specialized ones (Section A).

5 Experiments

We evaluate NBSR on six tasks of increasing structure. Architectures, data processing, hardware, and all hyperparameters are listed in Section F; extended analyses, routing traces, and additional figures are in Section D.

Toy illustration: sequential belief sharpening. On two Iris features, a hard-coded depth-2 path with a sigmoid-bounded mid expert and a Softplus leaf expert shows the predicted behavior (Fig. 3a): after depth 1 the decision regions are correct but faded (low precision); after depth 2 the precision rises and the boundaries become sharp and saturated, as Theorem 1 predicts.

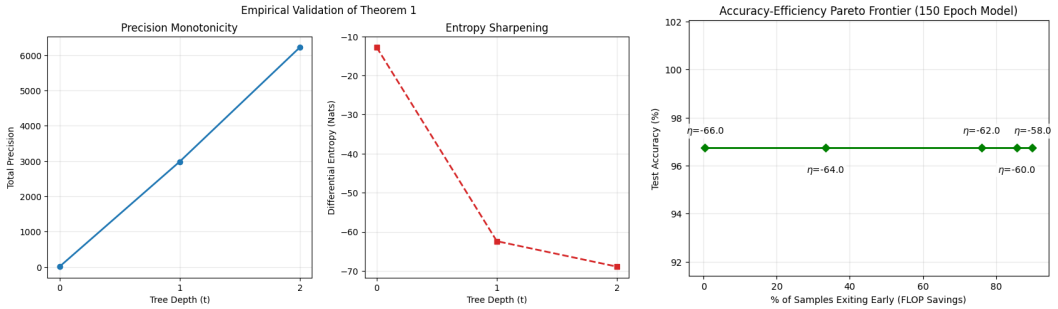
5.1 Visual categorization: CIFAR-10

We use an ImageNet-pretrained, unfrozen ResNet-18 as the oracle and a predefined two-level taxonomy (Animals/Vehicles, then five leaf experts: Pets, Wildlife, Birds, AirWater, Road), with each expert emitting 10-dimensional evidence. The baselines are a flat ResNet-18 and a sparse MoE with five experts, top-2 soft gating, and a load-balancing loss; all models share the backbone, augmentation, seed, and 150-epoch schedule.

Table 1: CIFAR-10 results. All models share the backbone, seed, and data pipeline; times are average per-epoch training/inference times.

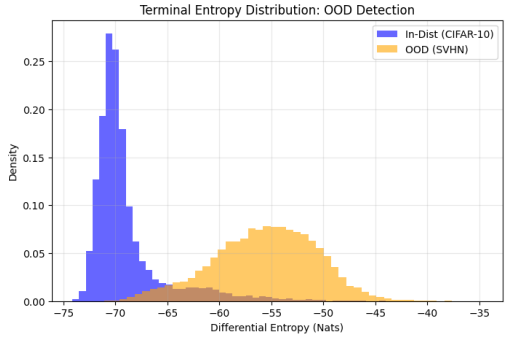
Model	Acc. (%) ↑	Time (train / test)	ECE ↓	Interpretability
Flat ResNet-18	96.60	53.1s / 5.0s	0.026	None (black box)
Sparse MoE (soft top-2)	96.43	54.6s / 5.2s	0.027	Gate weights only
NBSR (ours)	**96.74**	53.5s / 5.1s	**0.015**	Full routing audit trail

Results. NBSR attains the highest accuracy (96.74%) at essentially the cost of the flat network, and a markedly lower ECE (0.015 versus 0.026 and 0.027) computed on its Dirichlet means (Table 1); the accuracy differences are small, whereas the calibration gap is large. Belief sharpening follows

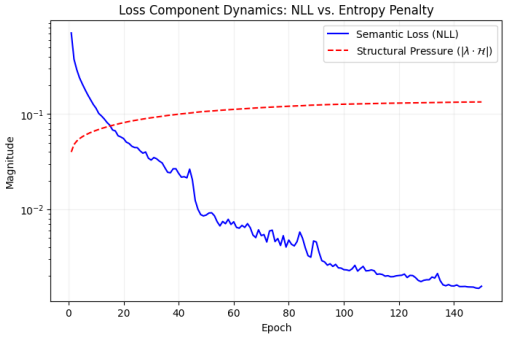


(a) Mean precision and entropy vs. depth

(b) Accuracy–efficiency frontier (early exit)



(c) Terminal entropy: CIFAR-10 (ID) vs. SVHN (OOD)



(d) Training losses: NLL vs. weighted entropy

Figure 2: NBSR on CIFAR-10. (a) Mean Dirichlet precision rises monotonically from 10 to over 6000 across depth, and mean entropy collapses (Theorem 1). (b) Sweeping $\eta \in [-66, -58]$: nearly 90% of test images exit at depth 1 with unchanged accuracy (96.74%). (c) SVHN (OOD) images retain higher terminal entropy than CIFAR-10 test images. (d) The NLL drives early feature learning; the entropy term dominates late training and sharpens the belief.

Theorem 1 (Fig. 2a). The depth-1 entropy test is an effective difficulty detector: relaxing η lets nearly 90% of images exit after one expert without accuracy loss (Fig. 2b). OOD inputs (SVHN) are assigned clearly higher terminal entropy (Fig. 2c). The two loss terms also separate in time (Fig. 2d): the NLL governs early feature learning and approaches its floor (about 10^{-3}) by epoch 100, after which the weighted entropy term $\lambda\mathcal{H}$ becomes the dominant component and drives *belief sharpening*, as Proposition 1 anticipates; $\lambda = 10^{-3}$ gave a stable balance. Soft top-2 routing, by contrast, gave the lowest accuracy and the most oscillatory training of the three models (Section D).

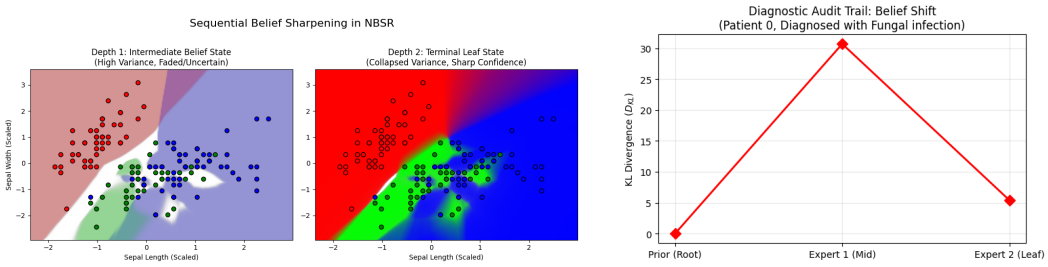
5.2 Structured Medical Diagnosis

We map 132 binary symptoms (with 5% injected noise) to 41 diseases through an MLP oracle and a triage graph (four physiological mid experts, then leaf experts), and compare against a flat MLP and XGBoost (Chen & Guestrin, 2016). Every prediction yields an *audit trail* of routing decisions and per-step belief shifts $\mathrm{KL}(\mathrm{Dir}(\boldsymbol{\alpha}_{t+1}) \| \mathrm{Dir}(\boldsymbol{\alpha}_t))$ (Fig. 3b). Because $\alpha_{T,y^*} = \alpha_{0,y^*} + \sum_t e_{t,y^*}$ is additive, input×gradient attribution decomposes exactly over the visited experts,

$$\mathcal{I}(x) = x \odot \frac{\partial \alpha_{T,y^*}}{\partial x} = x \odot \textstyle\sum_{t=0}^{T-1} \mathbf{J}_{\mathbf{h}_x}(x)^\top \nabla_{\mathbf{h}_x} e_{t,y^*}, \tag{4}$$

which gives patient-specific, per-expert rationales rather than XGBoost's population-level importances (Fig. 4).

Results. NBSR matches the flat MLP's accuracy (97.62%), while XGBoost reaches 100% on this tabular task (Table 2). The early exit at $\eta = -100$ halves the average depth without loss of accuracy. NBSR is, however, markedly less calibrated here (ECE 0.166–0.259): on this nearly deterministic symptom data, additive evidence grows the precision quickly and yields over-confident beliefs, so NBSR trades calibration for exact, sequential interpretability in this domain.



(a) Iris: depth 1 (left) vs. depth 2 (right)

(b) Diagnostic audit trail

Figure 3: (a) Toy experiment: colour hue is $\mathbb{E}[p_k]$ and saturation is the precision $\alpha_0^{(t)}$; the bounded mid expert gives a faded, uncertain hypothesis, and the leaf expert sharpens it. (b) The route of one patient with its sequential belief shifts (D_{KL}): Root $\rightarrow$ Dermatological expert $\rightarrow$ leaf, giving a confident diagnosis of fungal infection.

Table 2: Medical diagnosis (41 classes). NBSR is trained once; "Fast" uses $\eta = -100$ at test time.

Model	Acc.	Depth	Infer.	ECE
XGBoost	**100.0**	–	4.7ms	0.031
MLP	97.62	–	1.7ms	**0.016**
NBSR (Deep)	97.62	2.0	2.3ms	0.166
NBSR (Fast)	97.62	1.0	1.9ms	0.259

Table 3: Next-token prediction on a controlled grammar (65-token vocabulary). TPS: CPU tokens/s.

Model	PPL	Depth	TPS	ECE
Transformer	**13.43**	–	**62004**	0.011
Transf. MoE	**13.43**	2.0	54307	**0.008**
NBSR (Deep)	13.81	2.0	37376	0.022
NBSR (Fast)	15.37	1.0	29181	0.051

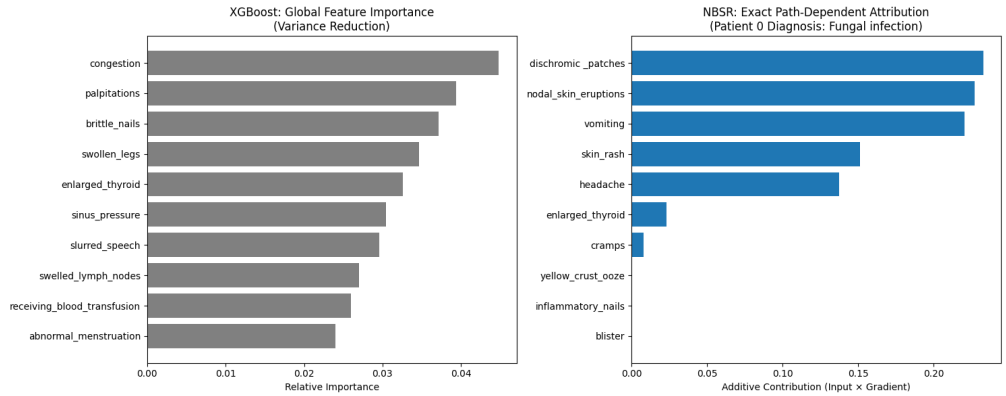


Figure 4: Comparison of Feature Attribution. (Left) XGBoost provides a global, population-level estimate of feature importance via variance reduction, which lacks patient-specific diagnostic relevance. (Right) NBSR utilizes path-dependent gradient attribution to extract the exact, localized biometric markers driving Patient 0's specific diagnosis (Fungal Infection).

5.3 Language Modeling

We replace the output head of a small causal Transformer with an NBSR graph whose levels mirror linguistic processing (syntactic vs. semantic, then function words, modifiers, abstract nouns, and concrete entities) and train it on a controlled grammar with 65 tokens. This corpus deliberately contains syntax without world knowledge. NBSR retains most of the backbone's capacity (perplexity 13.81 vs. 13.43), at a cost in throughput from dynamic routing masks and in calibration (Table 3). Its benefit is a readable reasoning trace: for "the dog is by the . . . ", the root sends all routing mass to the semantic expert, and the leaves extract evidence only for noun classes (abstract 275.5, concrete 320.4) and none for function words or modifiers, showing that it narrowed the prediction to a valid part of speech (Section D.3).

```
Prompt: "<bos> the dog is by the ..."   NBSR predicts: 'war'
 [Root router]   Syntactic: 0.00 | Semantic: 1.00
 [Leaf evidence] Function: 0.0 | Modifier: 0.0
                 Abstract: 275.5 | Concrete: 320.4
```

On an out-of-distribution prompt with an unknown token, all models give flat predictions ($\approx$6%) in this small vocabulary. NBSR additionally exposes the precision α_0, which supports evidential thresholding as an alternative to top-k decoding, although the 0% abstention rate at our threshold shows that this remains to be demonstrated at scale.

5.4 NBSR-MEM: PARTIALLY OBSERVABLE CONTROL

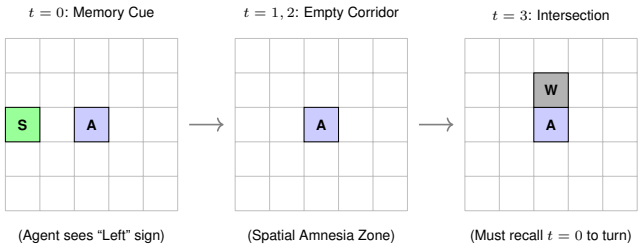


Figure 5: POMDP keyframes. At $t = 0$ a cue (S) sets the future turn; at $t = 1, 2$ the corridor is featureless ("spatial amnesia" for reactive policies); at $t = 3$ the agent meets a wall (W) and must recall the cue from memory.

Table 4: POMDP navigation (1000 held-out episodes). Abstention is the rate of safe halts on an unseen OOD hazard.

Policy	Success ↑	Avg. depth	OOD abstention ↑	Interpretability
CNN (reactive)	64.7%	–	0.0% (crashes)	Black-box
CNN-GRU (memory)	**100.0%**	–	0.0% (crashes)	Black-box
NBSR (reactive)	65.3%	1.18	**100.0%**	Full trace
NBSR-Mem (ours)	**100.0%**	1.00	**100.0%**	Full trace

An agent with a 5×5 local view must remember a cue seen at $t = 0$ in order to turn correctly at an intersection at $t = 3$, after an uninformative corridor. NBSR-Mem feeds CNN oracle features through a GRU memory before a Cruising/Evasion $\rightarrow$ action routing graph and is trained by behavioural cloning (Pomerleau, 1988). Recurrent evidential routing is prone to expert collapse and vanishing gradients (Shazeer et al., 2017; Pascanu et al., 2013); four interventions stabilized it: (i) a deterministic softmax router during training, so that backpropagation through time reaches the GRU smoothly; (ii) an auxiliary concept loss that maps corridors to the Cruising expert and intersections to the Evasion expert; (iii) a target-masked evidential regularizer (Sensoy et al., 2018) that penalizes evidence only on incorrect actions, which avoids a tug-of-war with the NLL (the plain entropy penalty of Eq. (3) caused optimization deadlock here); and (iv) LayerNorm and targeted weight decay on the backbone, which keep untrained input channels near zero (Section D.4). Reactive policies fail at intersections (about 65% success), whereas NBSR-Mem matches the CNN-GRU baseline (100%) (Table 4). Memory also reduces computation: the reactive NBSR is uncertain at intersections and routes to depth 2 there (average depth 1.18), whereas NBSR-Mem is confident enough to exit at depth 1 on every state (average depth 1.00). Unlike the opaque CNN-GRU, its decision at an intersection is fully traceable:

```
State: wall ahead, left cue seen at t=0.   NBSR-Mem action: Left
 [Root router]  Cruising: 0.00 | Evasion: 1.00
 [Depth 1]      Up: 0.0 | Down: 0.0 | Left: 28.3 | Right: 0.0
 [Depth 2]      Up: 0.0 | Down: 0.0 | Left: 75.0 | Right: 0.0
```

The root isolates the Evasion mode, and when traversal continues to depth 2 the leaf experts add evidence only for the correct action; evidence for every incorrect action stays at zero. With an "alien hazard" channel unseen during training, the softmax baselines always pick a movement and crash, whereas NBSR's total precision (8.39) stays below the abstention threshold (10.0) and the agent halts in 100% of trials, as Corollary 1 predicts.

5.5 AR-NBSR: COST-AWARE BAYESIAN EXPERIMENTAL DESIGN

In BOED, a design is chosen to maximize the expected information gain $\mathrm{EIG}(\xi) = \mathbb{E}_{p(e|\xi)}[\mathcal{H}(p(y)) - \mathcal{H}(p(y \mid e, \xi))]$ (Foster et al., 2019). In NBSR, a routing action is a design and the expert's evidence is the observation, so the realized gain $\mathcal{H}(\boldsymbol{\alpha}_t) - \mathcal{H}(\boldsymbol{\alpha}_{t+1})$ is available in closed form. We turn diagnosis into *active triage*: only demographics are free, and the 132 symptoms are split into five test panels with costs $c(v) \in \{10, 15, 20, 30, 50\}$. We train with

$$\mathcal{L}_{\mathrm{BOED}} = -\log \frac{\alpha_{T,y^*}}{\sum_k \alpha_{T,k}} + \lambda \sum_{v_t \in \mathcal{T}} \mathcal{H}(\boldsymbol{\alpha}_t) + \gamma \sum_{v_t \in \mathcal{T}} c(v_t), \tag{5}$$

where penalizing the area under the entropy curve rewards large early gains per unit cost. A static DAG over N panels and L steps would need $\mathcal{O}(N^L)$ nodes, so AR-NBSR instead reuses a single router over a flat pool of N experts in a recurrent loop, with state $\mathcal{S}_t = (\mathbf{h}_x, \boldsymbol{\alpha}_t, \mathbf{m}_t)$ and query history $\mathbf{m}_t$, stopping when $\mathcal{H}(\boldsymbol{\alpha}_t) < \eta$ (Fig. 6).

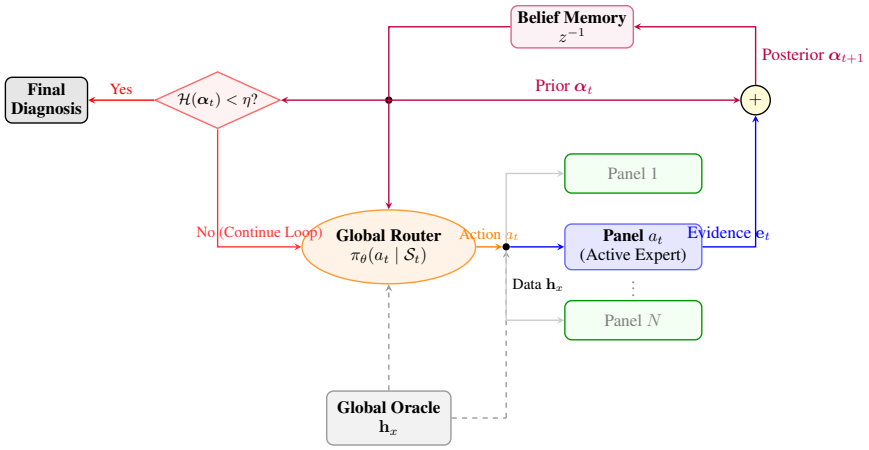
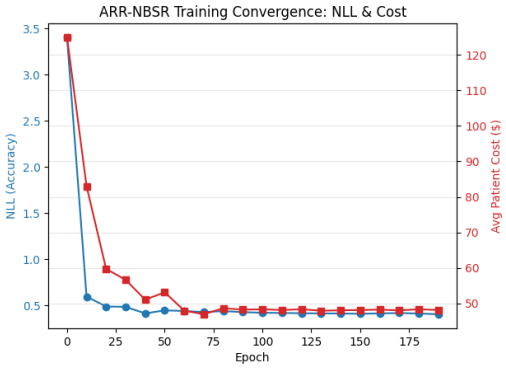


Figure 6: Illustration of the Autoregressive NBSR (AR-NBSR) framework. Unlike a static feed-forward DAG, the Global Triage Router iteratively queries a flat array of available diagnostic panels. The belief state $\boldsymbol{\alpha}_t$ acts as a recurrent memory buffer, denoted by the Unit Delay Operator (z^{-1}) from discrete-time signal processing, which caches the newly updated posterior to serve as the prior for the subsequent routing step. By sequentially accumulating evidence $\mathbf{e}_t$ via conjugate addition until the diagnostic entropy falls below the safety threshold η, this temporal feedback loop naturally averts the $\mathcal{O}(N^L)$ combinatorial explosion.

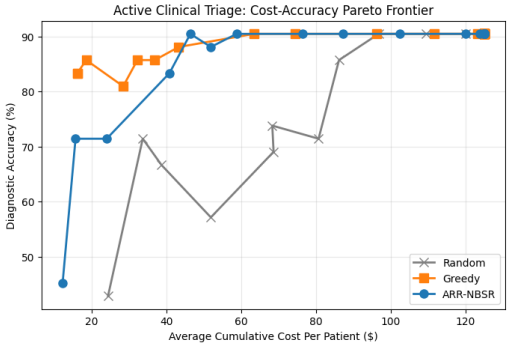
Results. Against random allocation and a greedy baseline that maximizes the immediate gain-to-cost ratio $\Delta\mathcal{H}_t/c(v)$, AR-NBSR yields the best cost–accuracy frontier (Fig. 7). The greedy policy often selects cheap, uninformative panels that require expensive follow-ups, while AR-NBSR sometimes chooses a more expensive panel early when it removes more downstream uncertainty. Training converges stably, with the average cost falling from \$125 to about \$48 (Fig. 7a). Per-patient traces show that entropy is reduced over several tests, each selected conditionally on the evidence gathered so far, rather than by a single decisive test, and that the loop stops as soon as the threshold is crossed (Fig. 7c; Section D.5).

6 DISCUSSION AND LIMITATIONS

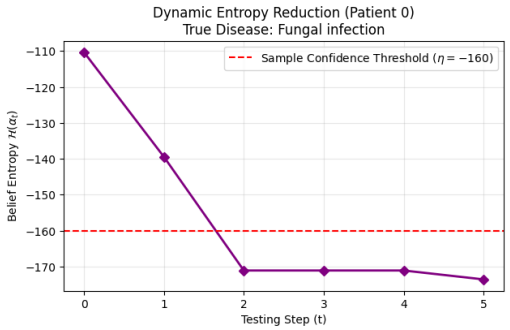
Design principles. Three choices are essential. First, the *persistent oracle* lets every node re-query the full input representation, avoiding progressive loss of information across layers (Shwartz-Ziv & Tishby, 2017); each expert acts as a task-specific probe of a shared semantic space. Second, the belief must *not be normalized*: the unnormalized precision α_0 carries the epistemic signal needed for abstention. Third, training uses *full depth but hard width*: disabling early exit during training avoids starving deeper experts of gradients, while hard routing prevents the blended, unspecialized experts of soft routing. NBSR can also be read as a policy in a Markov decision process over belief



(a) Training convergence



(b) Cost–accuracy frontier



(c) One patient's entropy trace

Figure 7: AR-NBSR for active clinical triage. (a) NLL (blue) and average cost per patient (red) fall together during training, from the \$125 budget ceiling to about \$48. (b) Frontier from sweeping 30 test-time thresholds η: AR-NBSR (blue) reaches its peak accuracy (about 90.5%) at about \$45–50 per patient, whereas greedy EIG (orange) needs about \$60–80 and random selection (grey) nearly the full budget. (c) Belief entropy decreases over several tests and the loop stops once it crosses η.

states, whose transitions are conjugate updates and whose reward is negative entropy minus cost; AR-NBSR makes this explicit (Section E).

Limitations. Our guarantees concern the precision and variance of the belief and hold under strictly positive evidence; Bayes-optimal recovery requires idealized capacity, $\lambda = 0$, and global optimization. The graph topologies are predefined, and the recurrent NBSR-Mem relies on auxiliary losses and a soft router during training. Scaling to larger vocabularies and real-world sequential tasks, improving calibration in high dimensions, and learning the topology (e.g., via Dirichlet-process routing) are important next directions.

7 Conclusion

NBSR turns neural prediction into a sequential, uncertainty-aware process of evidence acquisition: a hierarchy of experts repeatedly queries a persistent oracle, and their positive evidence accumulates as Dirichlet pseudo-counts under hard, belief-conditioned routing. This unifies conditional computation, evidential uncertainty, and sequential decision-making, with guaranteed precision growth, a bounded variance, and Bayes-optimal recovery in the ideal limit. Across vision, diagnosis, language, control, and active triage, NBSR is competitive in accuracy while providing early exit, abstention, cost-aware acquisition, and audit trails generated by the forward pass itself.

AI use statement

AI tools were used to aid and polish the writing and formatting of this paper. In addition, parts of the experimental Python code were developed with the assistance of AI. We have reviewed all AI-assisted text and code, and take full responsibility for the final content of this work.

Ethics statement

Our clinical experiments use a public, de-identified symptom–disease dataset with synthetic noise injection; no human subjects or private health records are involved. The diagnostic and triage results

are proofs of concept, not validated clinical tools: deploying NBSR for medical decisions would require prospective validation, calibration checks (we observe over-confidence on this dataset), and clinician oversight. NBSR's audit trails and abstention mechanism are intended to support, not replace, accountable human decision-making.

Reproducibility statement

Complete proofs are given in Section B, and the full training procedure in Algorithm 1 (Section A). Section F specifies, for every experiment, the hardware, data sources and preprocessing (including noise injection and data generation for the synthetic language and POMDP tasks), all network architectures, optimizers, schedules, temperatures, thresholds, loss weights, and random seeds. Code will be released upon publication.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, March 1994. ISSN 1045-9227. doi: 10.1109/72.279181. URL `https://doi.org/10.1109/72.279181`.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL `https://arxiv.org/abs/1308.3432`.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, 1 edition, 2006. ISBN 978-0-387-31073-2.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1613–1622. JMLR.org, 2015.

Elizabeth Baraff Bonawitz, Darlene Ferranti, Rebecca Saxe, Alison Gopnik, Andrew N. Meltzoff, James Woodward, and Laura E. Schulz. Just do it? investigating the gap between prediction and action in toddlers' causal inferences. *Cognition*, 115(1):104–117, 2010. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2009.12.001. URL `https://www.sciencedirect.com/science/article/pii/S0010027709002947`.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1 edition, 1984. doi: 10.1201/9781315139470.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL `https://doi.org/10.1145/2939672.2939785`.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989. doi: 10.1007/BF02551274.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers, 2020. URL `https://arxiv.org/abs/2003.07892`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, 2016.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, New York, 3 edition, 2013. ISBN 9780429113079. doi: 10.1201/b16018.

María F. Gil-Leyva, Antonio Lijoi, Ramsés H. Mena, and Igor Prünster. Markov stick-breaking processes, 2026. URL https://arxiv.org/abs/2601.16561.

Alex Graves. Adaptive computation time for recurrent neural networks, 2017. URL https://arxiv.org/abs/1603.08983.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic Neural Networks: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44 (11):7436–7456, November 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3117837. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3117837.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Tom Heskes. Bias/variance decompositions for likelihood-based estimators. *Neural Comput.*, 10 (6):1425–1433, August 1998. ISSN 0899-7667. doi: 10.1162/089976698300017232. URL https://doi.org/10.1162/089976698300017232.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://doi.org/10.1162/089976602760128018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL https://www.sciencedirect.com/science/article/pii/089360809190009T.

Yongchao Huang. Vjepa: Variational joint embedding predictive architectures as probabilistic world models, 2026. URL https://arxiv.org/abs/2601.14354.

Yongchao Huang. On the information bottleneck of VJEPA. *OpenReview*, May 2026. URL https://openreview.net/forum?id=S7hzZHbMwY.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), April 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL https://doi.org/10.1145/3054912.

Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001. doi: 10.1198/016214501750332758. URL https://doi.org/10.1198/016214501750332758.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pp. 1339–1344 vol.2, 1993. doi: 10.1109/IJCNN.1993.716791.

Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer Cham, 1 edition, 2016. ISBN 978-3-319-42335-7. doi: 10.1007/978-3-319-42337-1.

Daniel Jurafsky and James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 1st edition, 2000.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

Daniel Kahneman. *Thinking, Fast and Slow*. Penguin Books, London, 2011.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Yann LeCun. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015. doi: 10.1038/nature14539.

Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, September 2018. ISSN 0001-0782. doi: 10.1145/3233231. URL https://doi.org/10.1145/3233231.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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. URL https://openreview.net/forum?id=S1jE5L5gl.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. *Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1310–III–1318. JMLR.org, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Dean A. Pomerleau. Alvinn: an autonomous land vehicle in a neural network. In *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, NIPS'88, pp. 305–313, Cambridge, MA, USA, 1988. MIT Press.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1994. ISBN 978-0-471-61977-2. doi: 10.1002/9780470316887.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Howard Raiffa and Robert Schlaifer. *Applied Statistical Decision Theory*. Harvard University Press, Boston, 1961.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019. doi: 10.1038/s42256-019-0048-x.

Paola Sebastiani and Henry P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62 (1):145–157, 2000. ISSN 13697412, 14679868. URL http://www.jstor.org/stable/2680683.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3183–3193, Red Hook, NY, USA, 2018. Curran Associates Inc.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017. URL https://arxiv.org/abs/1703.00810.

A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2): 117–186, 1945. ISSN 00034851. URL http://www.jstor.org/stable/2235829.

S. R. Waterhouse and A. J. Robinson. Constructive algorithms for hierarchical mixtures of experts. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'95, pp. 584–590, Cambridge, MA, USA, 1995. MIT Press.

George Kingsley Zipf. *Human behavior and the principle of least effort: an introduction to human ecology*. Addison-Wesley, 1949.

A Method Details

A.1 Notation

To ensure clarity throughout the methodological and theoretical formulations, we summarize the key mathematical notations used in this paper in Table 5.

A.2 Training graph and gradient flow

A depth-2 computational graph illustrating the NBSR procedure (both the discrete forward execution path and the continuous backward gradient flow) is shown in Fig. 8.

A.3 The decision graph and the oracle

We formulate the sequential decision-making process as a traversal over a Directed Acyclic Graph[1] (DAG), denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ represents the set of computational *nodes* (*vertices*, acting as local experts and routing gates), and $\mathcal{E}$ represents the directed *edges* corresponding to routing actions. The process originates at a designated root node v_0, which acts purely as an initial routing gate without an evidence-extracting expert. From there, it sequentially traverses the graph. At each node $v_t \in \mathcal{V}$, a *conditional routing policy* determines the discrete outgoing edge $a_t \in \mathcal{A}(v_t)$, where $\mathcal{A}(v_t)$ is the action set at node v_t, to traverse, leading to the subsequent node v_{t+1}. This traversal governs the flow of information and concludes when a final prediction is rendered over the K-dimensional outcome space.

While the maximal topology of the graph $\mathcal{G}$ (i.e. the total number of available layers and expert nodes) is pre-specified as a fixed super-graph, the effective architecture is highly dynamic. The network autonomously performs implicit pruning during training; if an expert node fails to provide discriminative evidence, the upstream routers learn to assign near-zero probability to its incoming edges, effectively removing it from the computational flow. Further, the inference depth is dynamically truncated on a per-sample basis via the entropy-based early exiting mechanism.

By initializing a sufficiently large *super-graph*, we establish a high-capacity, low-bias foundation. Importantly, the aforementioned dynamic mechanisms act as adaptive regularizers: implicit routing pruning organically restricts the *effective width*, while entropy-based early exiting curtails the *effective depth*. Together, they dynamically rein in the variance, allowing the network to strike an optimal, sample-specific balance between representational capacity and generalization. Precise discussions

[1]While the primary formulation describes a strict feed-forward DAG, this topology can be temporally flattened into an *Autoregressive State Machine* by sharing a single router and expert pool across time. As we will demonstrate in Section 5.5, this cyclical extension prevents combinatorial explosion in long-horizon sequential planning tasks.

Table 5: Summary of Key Mathematical Notations

Notation	Description
Graph Topology & Traversal	
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Directed Acyclic Graph with computational nodes $\mathcal{V}$ and routing edges $\mathcal{E}$.
v_t, a_t	The node visited and the discrete routing action taken at step t.
$\mathcal{A}(v_t)$	The set of valid discrete actions (outgoing edges) originating from node v_t.
$\mathcal{T}$	A specific sampled trajectory (path) through the graph for a given input x.
L	The maximal pre-specified depth of the routing tree super-graph.
T	The terminal node (or terminal step) of a specific trajectory $\mathcal{T}$, where $T \leq L$.
$\mathcal{S}_t$	The joint decision state at step t, incorporating context, belief, and memory history.
Data & Representations	
x, y^*	The input data instance and its corresponding ground-truth target class.
K	The dimensionality of the outcome space (e.g. total number of categories).
$\mathbf{h}_x$	The Global Knowledge Oracle (dense feature vector in $\mathbb{R}^d$) generated by backbone f_ϕ.
$\boldsymbol{\varepsilon}_t$	Optional contextual side information available at step t.
$\mathbf{m}_t$	The recurrent memory buffer (e.g. GRU hidden state) encoding past trajectory observations.
Neural Components	
f_ϕ	The shared global feature backbone, parameterized by ϕ, mapping input x to oracle state $\mathbf{h}_x$.
$\text{MLP}_{\theta_{v_t}}$	The Router network at node v_t, parameterized by θ_{v_t}, outputting routing logits $\mathbf{z}_t$.
f_{v_t}	The Expert network at node v_t, parameterized by weights $\mathbf{W}_{v_t}$ and bias $\mathbf{b}_{v_t}$.
$\mathbf{e}_t$	The strictly positive, continuous evidence vector ($\in \mathbb{R}_+^K$) extracted by the expert f_{v_t}.
τ	The Gumbel-Softmax temperature controlling the continuous relaxation of the router.
Θ	The complete set of all trainable parameters within the end-to-end framework.
Bayesian State, Uncertainty & Optimization	
$\boldsymbol{\alpha}_t$	The Dirichlet concentration parameter vector (belief state) at step t, where $\boldsymbol{\alpha}_t \in \mathbb{R}_+^K$.
$\alpha_0^{(t)}$	The precision (total sum of concentration parameters) of the belief state: $\sum_k \alpha_{t,k}$.
$\mathbf{p}$	The Categorical probability distribution over the K classes.
u	Epistemic uncertainty derived from Subjective Logic ($u = K/\alpha_0^{(t)}$), utilized for OOD abstention.
$\mathcal{H}(\boldsymbol{\alpha}_t)$	The differential entropy of the Dirichlet belief state.
D_{KL}	The Kullback-Leibler divergence representing the sequential belief shift ($D_{\text{KL}}(\boldsymbol{\alpha}_{t+1} \parallel \boldsymbol{\alpha}_t)$).
η	The entropy confidence threshold utilized for dynamic early exiting.
$c(v_t), \gamma$	The asymmetric measurement cost of expert v_t, and its associated budgetary penalty scalar.
λ	The tunable hyperparameter scaling the temporal entropy penalty in the loss function.

regarding the bias-variance trade-off, induced by the graph topology, are presented in Section 4 and below.

To provide a stable informational foundation for the tree, the input x (e.g. visual or tabular data) is embedded *once* into a dense feature space to create a *Global Knowledge Oracle*, denoted as $\mathbf{h}_x$. Formally, given a shared neural backbone f_ϕ parameterized by ϕ, the oracle is computed as:

$$\mathbf{h}_x = f_\phi(x) \tag{6}$$

This shared representation acts as a persistent knowledge base that encapsulates the totality of the external data available for a given instance. For example, for visual tasks, we employ a truncated ResNet-18 architecture as f_ϕ, with the final fully connected classification layer removed, to extract a dense, continuous feature representation $\mathbf{h}_x \in \mathbb{R}^d$, where e.g. $d = 256$. Rather than transforming the data sequentially layer-by-layer, this global oracle $\mathbf{h}_x$ is broadcast to the entire graph, allowing all active nodes in the routing tree to query it directly to forge node-specific, local knowledge. Note that, unlike pixel-level models such as random forest or CART decision trees[2] (Breiman et al., 1984), which look at one raw feature at each hierarchical splitting node, the experts in NBSR partition the semantic space (e.g. "Setosa vs. Non-Setosa" in Iris classification), not the raw input space.

[2]A comparison between NBSR and decision trees is made in Appendix.C.

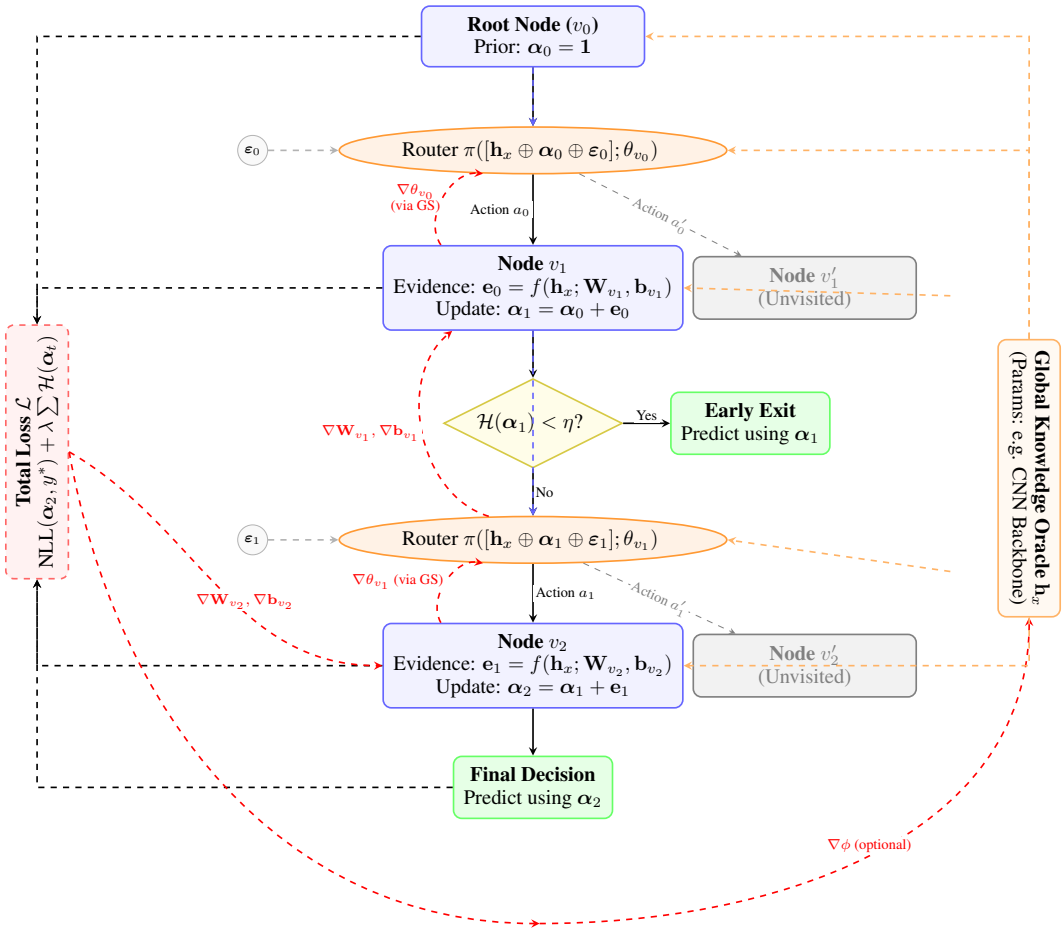


Figure 8: Computational graph of the NBSR process. The total number of sequential layers (maximum tree depth) and the number of expert nodes at each layer (branching width) are predefined hyperparameters that establish the maximal capacity of the super-graph. The **Global Knowledge Oracle** is positioned in the outer loop, while the **Total Loss** resides in the inner loop. The oracle provides a persistent feature set $\mathbf{h}_x$ that is queried by both routers and experts. **Forward Pass:** Path selection is conditioned on current belief and oracle data. **Backward Pass:** Gumbel-Softmax (GS) allows end-to-end gradient updates.

A.4 Router and Expert Details

At step t, located at node v_t, the model must select a single discrete branch $a_t \in \mathcal{A}(v_t)$ to traverse deeper into the tree. The routing decision is conditioned on both the input features and the current shape of the model's uncertainty.

The router is instantiated as a *Multi-Layer Perceptron*[3] (MLP). It accepts as input a concatenation of the global feature vector $\mathbf{h}_x$ with the current belief state $\boldsymbol{\alpha}_t$, alongside any optional side information $\boldsymbol{\varepsilon}_t$:

$$\mathbf{z}_t = \mathrm{MLP}_{\theta_{v_t}}([\mathbf{h}_x \oplus \boldsymbol{\alpha}_t \oplus \boldsymbol{\varepsilon}_t]) \tag{7}$$

From a practical implementation standpoint, the router network at any given node v_t must have a statically defined output dimension. In standard deep learning frameworks such as PyTorch (Paszke et al., 2019) or TensorFlow (Abadi et al., 2015), the final linear layer of this MLP is hard-coded at initialization to output exactly $|\mathcal{A}(v_t)|$ routing logits $\mathbf{z}_t$, corresponding directly to the total number of possible discrete actions (child nodes) branching from v_t.

[3]The router architecture can be chosen as any reasonable network; here we choose MLP as an example.

Example 1 (MLP Router Architecture for CIFAR-10 Classification)**.** *For a typical visual categorization task with a K-dimensional outcome space and a convolutional backbone extracting d-dimensional features, the router is parameterized as follows:*

- *Input Dimension: $d + K + |\boldsymbol{\varepsilon}_t|$ (e.g. $256 + 10 = 266$ for standard CIFAR-10 without side information).*
- *Hidden Layer: linear projection to 128 dimensions, followed by a ReLU activation.*
- *Output Layer: linear projection to $|\mathcal{A}(v_t)|$ dimensions (the number of outgoing branches from node v_t).*

Once a target node v_{t+1} is activated, its resident "expert" network extracts local evidence from the input to update the belief state. Conceptually, this process treats the global feature representation as a *persistent knowledge base*, driven by three interacting components:

- *The Global (Knowledge) Oracle ($\mathbf{h}_x$):* serves as the static, comprehensive reference of the input instance, as defined in Section 3.
- *The Context-Aware Query ($\mathbf{z}_t$):* at step t, the router asks: "based on what I already know (i.e. the current state $\boldsymbol{\alpha}_t$) and what the oracle holds ($\mathbf{h}_x$), which expert should I consult next?"
- *The Information Extractor ($f_{v_{t+1}}$):* the chosen expert network $f_{v_{t+1}}$ acts as a specific, selective querying function[4] $\mathbf{e}_t = f(\mathbf{h}_x; \mathbf{W}_{v_{t+1}}, \mathbf{b}_{v_{t+1}})$. It interrogates the oracle $\mathbf{h}_x$ from a highly specialized perspective (e.g. "look specifically for bird-like features") and extracts only the relevant localized evidence $\mathbf{e}_t$.

Mathematically, the evidence extractor typically consists of a *linear* mapping from the shared feature vector $\mathbf{h}_x$ to the K-dimensional outcome space. Here, the trainable parameters $\mathbf{W}_{v_{t+1}}$ and $\mathbf{b}_{v_{t+1}}$ dictate precisely what information to extract from the global oracle given the specific context of the current node. To satisfy the strict positivity requirement of the Dirichlet concentration parameters, the output must be passed through a bounding activation function:

The choice of this activation function dictates the *epistemic capacity* of the expert. For terminal leaf experts, this is typically the unbounded *Softplus* function[5] ($\ln(1 + e^x)$), which allows the network to inject massive evidence to crystallize a final, highly confident decision. However, to mathematically enforce a hierarchical taxonomy, intermediate routing nodes can utilize a scaled *Sigmoid* activation. This places a strict upper bound on the evidence an intermediate node can extract, acting as an *epistemic confidence budget* that ensures broad, early-stage hypotheses remain appropriately hesitant (a dynamic visually demonstrated in our toy experiment in Section 5).

Example 2 (CIFAR-10 Evidence Extractor)**.** *Following the previous visual categorization setup in Example 1, the specific architecture for the local evidence extractor at each terminal node v_{t+1} is configured as follows:*

- *Input layer: receives the d-dimensional global feature vector $\mathbf{h}_x$ (e.g. $d = 256$ from the truncated ResNet-18 backbone).*
- *Transformation: a single linear fully-connected layer, parameterized by a weight matrix $\mathbf{W}_{v_{t+1}} \in \mathbb{R}^{K \times d}$ and bias $\mathbf{b}_{v_{t+1}} \in \mathbb{R}^K$.*

[4] **Static vs. Dynamic Experts:** our framework currently employs *Static Experts* with implicit context. Because the router already used the belief state $\boldsymbol{\alpha}_t$ to select node v_{t+1}, the simple act of arriving at this node is implicitly conditioned on $\boldsymbol{\alpha}_t$. The matrix $\mathbf{W}_{v_{t+1}}$ represents a fixed, specialized filter; it does not need explicit knowledge of $\boldsymbol{\alpha}_t$ to extract its specific feature. This static design ensures faster inference, easier training, and conceptual clarity. An alternative is a *Dynamic Expert* framework formulated as $f(\mathbf{h}_x, \boldsymbol{\alpha}_t; \mathbf{W}, \mathbf{b})$. In this paradigm, $\boldsymbol{\alpha}_t$ acts as an explicit query in an *attention* mechanism, while $\mathbf{h}_x$ acts as the key/value, allowing the expert to dynamically shift its weights based on the model's exact current uncertainty. We leave the exploration of dynamic attention-based experts to future work.

[5] The Softplus function is a smooth, differentiable activation function defined as $\text{Softplus}(x) = \ln(1 + e^x)$, acting as a continuous approximation of the ReLU activation function. It produces positive outputs ranging from 0 to ∞, commonly used to constrain output values to be positive in neural networks, such as in regression tasks, and has a derivative equal to the sigmoid function.

- *Activation: an unbounded Softplus activation function is applied to the raw logits to guarantee mapping into the strictly positive domain ($\mathbb{R}_+^K$).*

- *Output dimension: a K-dimensional evidence vector $\mathbf{e}_t$ (e.g. $K = 10$ representing the isolated evidence for each categorical outcome).*

A.5 Training dynamics and algorithm

Training the tree structure admits a highly efficient, sample-specific gradient flow. Because of the discrete routing decisions made during the forward pass, gradients only propagate through the active trajectory $\mathcal{T}$. Consequently, *unvisited experts* receive zero gradients and remain unmodified by that specific instance. Conversely, *active experts* along the traversed path are directly optimized to extract more discriminative evidence from the global oracle. Simultaneously, the *active routers* learn to adjust their conditional routing probabilities via the continuous backward pass of the Gumbel-Softmax estimator; if a chosen path yields a high loss, the router penalizes the corresponding action logit, which inherently boosts the probability of selecting an alternative, unvisited branch in the future. This dynamic interplay of selective evidence extraction and adaptive routing is precisely what allows the tree to autonomously specialize, naturally evolving the computational graph into a highly structured, conditional *Mixture-of-Experts* (MoE).

We present the end-to-end training process in Algo.1, which explicitly details both the forward inference trajectory and the backward gradient flow, demonstrating how conditional execution dynamically restricts the computational cost. Importantly, the backward pass highlights that the tree organically prunes its gradient updates: only the routers and experts along the actively sampled trajectory $\mathcal{T}$ receive learning signals for any given instance.

A.6 Computational complexity

The resource-rationality of the NBSR framework stems directly from its decoupling of the structural graph capacity (which is exponentially large) from the active computational path (which is strictly linear and dynamically truncated). We formalize this across time and space domains:

- *Inference Time Complexity (FLOPs):* let $\mathcal{O}(C_{\text{backbone}})$ denote the complexity of the global feature extractor. The routing tree has a maximum depth L and branching factor $\bar{W}$. Because of the hard-routing `argmax` step during inference (or the STE during training), the input traverses *exactly one path*. The computational cost at any single active node consists of the Router MLP operations $\mathcal{O}(d \cdot d_h + d_h \cdot \bar{W})$ and the Expert linear projection $\mathcal{O}(d \cdot K)$, where d is the oracle dimension and d_h is the router's hidden size.

  Importantly, because of the entropy threshold η, the actual number of steps evaluated is a sample-dependent random variable $T(x) \leq L$. The expected total inference complexity is therefore:

$$\mathcal{O}\Big(C_{\text{backbone}} + \mathbb{E}[T(x)] \cdot (d \cdot d_h + d \cdot K)\Big) \tag{8}$$

  Because d, K, and d_h are typically small (e.g. $d = 256, K = 10$), the routing overhead $\mathbb{E}[T(x)]$ is vastly marginalized by the backbone cost. Further, because unambiguous samples trigger $T(x) \ll L$, the model actively *saves* computational cycles compared to static deep networks that must execute all layers for all inputs.

- *Space and Parameter Complexity:* a fully populated super-graph of depth L contains $\mathcal{O}(\bar{W}^L)$ discrete nodes. The total parameter footprint of the graph scales exponentially: $\mathcal{O}\big(\bar{W}^L \cdot d(d_h + K)\big)$. While this demands sufficient GPU VRAM for storage during training, the *active memory footprint* during a forward pass breaks this exponential scaling. Because unvisited nodes are bypassed via conditional execution, the active activation memory scales strictly linearly with the traversal depth, bounded by $\mathcal{O}(L \cdot (d_h + K))$. This makes the NBSR framework highly scalable to wide hierarchies during inference without causing combinatorial explosions in activation memory.

Algorithm 1 End-to-End Training of Neural Bayesian Sequential Routing Tree

Require: Training dataset $\mathcal{D}$, initialized parameters $\Theta = \{\phi, \theta, \mathbf{W}, \mathbf{b}\}$, learning rate ρ
Require: Max depth L, branching factor $\bar{W}$ (action space), confidence threshold η ($\eta = -\infty$ disables early exit during training), entropy penalty λ

while not converged **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{D}$
  Initialize batch loss $\mathcal{R} \leftarrow 0$
  for each instance $(x, y^*) \in \mathcal{B}$ **do**
    // 1. Forward Pass & Trajectory Sampling
    $t \leftarrow 0$, $v_t \leftarrow v_0$ (Root), $\boldsymbol{\alpha}_0 \leftarrow \mathbf{1} \in \mathbb{R}^K$
    $\mathcal{T}^{(x)} \leftarrow \{v_0\}$
    $\mathbf{h}_x \leftarrow f_\phi(x)$ {Global Knowledge Oracle}
    while $t < L$ **do**
      if $\mathcal{H}(\boldsymbol{\alpha}_t) < \eta$ **then**
        break {Uncertainty sufficiently resolved; exit early}
      end if
      $\mathbf{z}_t \leftarrow \text{MLP}_{\theta_{v_t}}([\mathbf{h}_x \oplus \boldsymbol{\alpha}_t \oplus \boldsymbol{\varepsilon}_t])$
      $a_t \sim \text{Gumbel-Softmax}(\mathbf{z}_t)$ {Sample discrete path via Straight-Through Estimator}
      $v_{t+1} \leftarrow \text{TargetNode}(v_t, a_t)$
      $\mathcal{T}^{(x)} \leftarrow \mathcal{T}^{(x)} \cup \{v_{t+1}\}$
      $\mathbf{e}_t \leftarrow \text{Activation}(\mathbf{W}_{v_{t+1}} \mathbf{h}_x + \mathbf{b}_{v_{t+1}})$
      $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \mathbf{e}_t$ {Bayesian Conjugate Update}
      $t \leftarrow t + 1$
    end while
    $T \leftarrow t$ {Record terminal step}
    // 2. Instance Loss Computation
    $\mathcal{L}_{\text{NLL}}^{(x)} \leftarrow -\log\left(\frac{\alpha_{T,y^*}}{\sum_{k=1}^K \alpha_{T,k}}\right)$
    $\mathcal{L}_{\text{Reg}}^{(x)} \leftarrow \lambda \sum_{i=0}^{T} \mathcal{H}(\boldsymbol{\alpha}_i)$
    $\mathcal{R} \leftarrow \mathcal{R} + \left(\mathcal{L}_{\text{NLL}}^{(x)} + \mathcal{L}_{\text{Reg}}^{(x)}\right)$
  end for
  // 3. Backward Pass (Gradient Computation)
  $\mathcal{R} \leftarrow \frac{1}{|\mathcal{B}|}\mathcal{R}$ {Average empirical risk over minibatch}
  Compute gradients $\nabla_\Theta \mathcal{R}$ via backpropagation
  {**Note:** For any instance x, $\nabla_{\mathbf{W}_v} \mathcal{R} = 0$ and $\nabla_{\theta_v} \mathcal{R} = 0$ for all nodes $v \notin \mathcal{T}^{(x)}$}
  // 4. Parameter Update
  $\Theta \leftarrow \text{OptimizerStep}(\Theta, \nabla_\Theta \mathcal{R}, \rho)$
end while
Return: Optimized parameters Θ^*

A.7 Practical implementation: backbone freezing and two-stage training

A critical architectural consideration is whether to train the backbone parameters ϕ end-to-end or keep them frozen as a static feature extractor. A permanently frozen backbone offers significant computational advantages: first, it saves training compute and memory[6]. Further, freezing ϕ drastically improves early-stage training stability. Because Gumbel-Softmax routing is inherently highly stochastic in early epochs, an actively updating backbone causes the feature space $\mathbf{h}_x$ to shift wildly, impeding the local experts' ability to learn reliable extraction filters. Utilizing frozen foundation models (e.g. DINOv2 (Oquab et al., 2024) or CLIP (Radford et al., 2021)) provides an exceptionally rich, anchored representation space for the routers to latch onto immediately; NBSR tree can focus entirely on learning the logical reasoning and routing rather than basic feature edge-detection.

However, a strictly frozen oracle imposes a *representation bottleneck* (Huang, May 2026). Pretrained features are optimal for their source tasks (e.g. general ImageNet categorization) and may lack the specific, fine-grained granularity required by downstream experts for specialized domains (e.g. medical diagnostics). Further, freezing prevents *synergistic adaptation*, forcing the tree to

[6]Training a complex backbone, e.g. ResNet-18 (He et al., 2016) or ViT (Dosovitskiy et al., 2021), requires storing intermediate activations or compute gradients for millions of parameters during the backward pass; avoiding training the backbone allows the routing tree to be trained on smaller hardware constraints with vastly larger batch sizes.

adapt to an arbitrary manifold rather than allowing the backbone to organically organize its latent space to support the routing hierarchy, which can artificially cap maximal accuracy.

To reconcile these trade-offs, we recommend a *Two-Stage Training Strategy* (Backbone Warm-up). In *Stage 1*, the pre-trained global oracle ϕ is *frozen*, and only the routers (θ_{v_t}) and experts ($\mathbf{W}_{v_t}, \mathbf{b}_{v_t}$) are trained for the first majority of the epochs (e.g. 70%). This allows the tree topology to discover stable routing paths and reliable evidence extractors without a shifting foundation. In *Stage 2*, the backbone ϕ is *unfrozen*, and the entire system is fine-tuned end-to-end using a significantly reduced learning rate (e.g. scaled by 10^{-1} or 10^{-2}). This stage allows the backbone to subtly adjust its feature space to perfectly serve the highly specialized experts that emerged during the warm-up phase, maximizing capacity without sacrificing stability.

A.8 Topological bias–variance trade-off

The structural configuration of the pre-specified super-graph natively induces a classic *bias-variance trade-off*. To formalize this, consider the expected NLL risk of our model over a *target conditional distribution* $P^*(y|x)$. Let $P_{\mathcal{G}}(y|x;\mathcal{D})$ denote the *predictive distribution* produced by our graph $\mathcal{G}$ trained on dataset $\mathcal{D}$, and let $\bar{P}_{\mathcal{G}}(y|x) = \mathbb{E}_{\mathcal{D}}[P_{\mathcal{G}}(y|x;\mathcal{D})]$ be the *expected prediction* over all possible datasets. Following the standard generalized bias-variance decomposition for cross-entropy loss, the expected risk can be expressed as (Heskes, 1998):

$$\mathbb{E}_{\mathcal{D}}\Big[\mathbb{E}_{y\sim P^*(\cdot|x)}\big[-\log P_{\mathcal{G}}(y|x;\mathcal{D})\big]\Big] \approx \underbrace{\mathcal{H}(P^*)}_{\text{Irreducible Noise}} + \underbrace{D_{\mathrm{KL}}\big(P^*\|\bar{P}_{\mathcal{G}}\big)}_{\text{Bias}} + \underbrace{\mathbb{E}_{\mathcal{D}}\big[D_{\mathrm{KL}}\big(\bar{P}_{\mathcal{G}}\|P_{\mathcal{G}}(\cdot;\mathcal{D})\big)\big]}_{\text{Variance}} \tag{9}$$

where $\mathcal{H}$ is the Shannon entropy and D_{KL} denotes the Kullback-Leibler divergence. The dimensions of our routing tree, namely its maximal depth L and its average branching width $\bar{W}$, govern the tension between the Bias and Variance terms:

- *Width (Branching Factor):* increasing the number of outgoing edges $\bar{W} = \mathbb{E}[|\mathcal{A}(v_t)|]$ allows for a finer partition of the input feature space. This high expressivity reduces the *Bias* term, as the model can dedicate highly specialized local experts to distinct sub-populations of the data. However, excessive width leads to severe data fragmentation. Assuming a balanced tree, the expected number of training samples reaching a node at depth l scales proportionally to $|\mathcal{D}|/\bar{W}^l$. As the sample size per node plummets, the local parameter estimates ($\mathbf{W}_v, \mathbf{b}_v$) become highly sensitive to noise in the specific training set $\mathcal{D}$, drastically increasing the *Variance* term and risking localized overfitting.

- *Depth (Sequential Steps):* deeper graphs ($L \to$ large) allow for prolonged evidence accumulation and complex hierarchical reasoning, further minimizing the *Bias* by approximating highly non-linear decision boundaries. Yet, excessive depth exponentially exacerbates the *Variance* through two mechanisms: compounding routing stochasticity along the trajectory, and forcing the fragmentation of data across $\bar{W}^L$ terminal leaf nodes.

Adaptive Regularization via the Belief State: if the tree topology were fixed and fully traversed for every input, minimizing the expected risk would require an arduous manual search for the optimal hyper-parameters $(L, \bar{W})$. Our framework circumvents this via the dynamically updated Dirichlet belief state $\boldsymbol{\alpha}_t$.

By imposing an intermediate entropy penalty $\mathcal{H}(\boldsymbol{\alpha}_t)$ and utilizing early exiting thresholds η, the network replaces the global structural constants L and $\bar{W}$ with sample-dependent variables $L(x)$ and $W(x)$. Easy, unambiguous samples are routed through shallow, highly populated paths ($L(x)$ is small, maximizing local data density and minimizing variance), while complex samples are permitted to traverse deeper, specialized sub-graphs ($L(x)$ is large, minimizing bias). Consequently, the framework autonomously calibrates the bias-variance trade-off on a per-instance basis, maximizing representational capacity without sacrificing estimator stability.

A.9 DECOUPLING OF TRAINING AND INFERENCE

Our NBSR framework explicitly decouples the training and inference regimes to simultaneously maximize *skill acquisition* and *computational efficiency*. In practice, we must balance two critical geometric aspects of the routing tree:

Depth and the "Lazy Optimization" Problem. During training, ideally the dynamic early-exiting mechanism (controlled by the entropy threshold η) must be intentionally disabled, forcing every sample to traverse the full depth of the graph. Neural networks are inherently "lazy" optimizers; they seek the path of least mathematical resistance to minimize the loss function. If early exiting were permitted during the learning phase, intermediate experts would quickly learn *just enough* rudimentary features to push the differential entropy slightly below η, halting the forward pass prematurely. Consequently, deeper leaf experts would experience *gradient starvation*, rendering the extended architecture functionally dead. By forcing full-depth traversals during training, we guarantee that every level of the graph receives the necessary gradients to acquire specialized skills. At inference time, the η threshold is activated, allowing the model to resource-rationally reap the computational savings of these rigorously acquired skills.

Width and the "Group Project" Problem. Conversely, while utilizing the full *depth* is essential during training, utilizing the full *width* (i.e. soft routing) is catastrophic. If the framework utilized standard continuous relaxation where an input traverses every branch and the final update is a probability-weighted average of all terminal leaf experts, the model would fall into *representation collapse* (Shazeer et al., 2017). This dynamic is analogous to a "group project" where the final grade is based on a blended average: because the network relies on the ensemble to minimize the loss, individual experts fail to specialize, instead learning generic, overlapping, "blurry" features. NBSR resolves this via the Gumbel-Softmax Straight-Through Estimator (STE). By enforcing hard discrete routing during the forward pass, the selected expert is forced to assume 100% responsibility for the prediction, mathematically guaranteeing absolute specialization. During the backward pass, the continuous relaxation allows the routing policy to evaluate counterfactual probabilities and learn optimal pruning strategies without ever incurring the computational cost of the unselected experts.

B PROOFS

B.1 PROOF OF THEOREM 1

By definition, the initial state is $\boldsymbol{\alpha}_0 = \mathbf{1}$, so $\alpha_0^{(0)} = K$. At each step t, the extracted evidence is computed as $\mathbf{e}_t = \text{Activation}(\mathbf{W}_{v_{t+1}}\mathbf{h}_x + \mathbf{b}_{v_{t+1}})$. Because the range of our allowable activation functions, whether the unbounded Softplus $(0, \infty)$ for terminal leaves or the bounded scaled Sigmoid $(0, C)$ for intermediate experts, is strictly positive, every extracted element satisfies $e_{t,k} > 0$. The conjugate update rule $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t$ dictates that the new total precision is $\alpha_0^{(t+1)} = \alpha_0^{(t)} + \sum_k e_{t,k}$. Since $\sum_k e_{t,k} > 0$, it strictly follows that $\alpha_0^{(t+1)} > \alpha_0^{(t)}$.

Further, the variance of the k-th marginal is $\text{Var}[p_k^{(t)}] = \frac{\alpha_{t,k}(\alpha_0^{(t)} - \alpha_{t,k})}{(\alpha_0^{(t)})^2(\alpha_0^{(t)}+1)}$. The numerator represents the product of two terms, $\alpha_{t,k}$ and $(\alpha_0^{(t)} - \alpha_{t,k})$, whose sum is fixed at $\alpha_0^{(t)}$. By the Arithmetic Mean-Geometric Mean (AM-GM) inequality, this product is maximized when the two terms are equal (i.e. when $\alpha_{t,k} = \frac{1}{2}\alpha_0^{(t)}$). This yields the strict upper bound $\alpha_{t,k}(\alpha_0^{(t)} - \alpha_{t,k}) \leq \frac{1}{4}(\alpha_0^{(t)})^2$. Applying this bound to the variance equation, we see the variance is strictly upper-bounded by $\frac{1}{4(\alpha_0^{(t)}+1)}$. Since $\alpha_0^{(t)}$ strictly increases at every step, this proves that the variance inevitably collapses as evidence accumulates.

B.2 PROOF OF THEOREM 2

Based on the Universal Approximation Theorem (Cybenko, 1989; Hornik, 1991), the assumption of infinite functional capacity is theoretically satisfied if the neural networks are constructed with either an arbitrarily large number of hidden units (infinite width) or an arbitrarily large number of

hidden layers (infinite depth), paired with a non-polynomial continuous activation function (such as ReLU). Under these conditions, the networks can approximate any continuous Borel measurable function on a compact domain to an arbitrary degree of precision.

Setting the entropy penalty coefficient $\lambda \to 0$ for the analysis of the primary objective, the regularization term vanishes. From the properties of the Dirichlet distribution, the expected marginal probability for the target class y^* is exactly $\mathbb{E}[p_{y^*} \mid \boldsymbol{\alpha}_T] = \frac{\alpha_{T,y^*}}{\sum_{k=1}^{K} \alpha_{T,k}}$. Substituting this into the primary loss term in Eq.3 reduces it directly to the Negative Log-Likelihood (NLL) of the model's expected prediction for a single data instance (x, y^*): $\mathcal{L}_{\text{NLL}} = -\log \mathbb{E}[p_{y^*} \mid \boldsymbol{\alpha}_T]$.

To demonstrate why minimizing this NLL yields the true posterior, we define the model's predicted conditional distribution for any class y given input x as $q(y \mid x) = \mathbb{E}[p_y \mid \boldsymbol{\alpha}_T(x)]$. The instance-wise NLL is therefore simply $-\log q(y^* \mid x)$. During end-to-end training, minimizing the empirical risk over a dataset corresponds to minimizing the *expected NLL* over the true joint data distribution $P_{\text{data}}(x, y)$. This expected loss is mathematically equivalent to the *cross-entropy* between the true data distribution and the model's predicted distribution, which can be decomposed into the Shannon entropy of the data and the Kullback-Leibler (KL) divergence:

$$\begin{aligned}
\mathbb{E}_{x,y\sim P_{\text{data}}}\left[\mathcal{L}_{\text{NLL}}\right] &= \mathbb{E}_{x,y\sim P_{\text{data}}}\left[-\log q(y \mid x)\right] \\
&= \mathbb{E}_x\left[\sum_{k=1}^{K} P(y=k \mid x)(-\log q(y=k \mid x))\right] \\
&= \mathbb{E}_x\left[\mathcal{H}(P(y \mid x)) + D_{\text{KL}}(P(y \mid x) \parallel q(y \mid x))\right]
\end{aligned}$$

Because the true data entropy $\mathcal{H}(P(y \mid x))$ is an irreducible constant with respect to the network parameters, *minimizing the expected training objective is mathematically equivalent to minimizing the KL divergence*. A fundamental property of KL divergence is Gibbs' inequality, which states $D_{\text{KL}}(P \parallel q) \geq 0$, with equality holding if and only if $P(y \mid x) = q(y \mid x)$ almost everywhere.

Since we assume the hypothesis class has infinite functional capacity, the network is not constrained by approximation error. Therefore, achieving the global minimum of the loss forces $D_{\text{KL}} \to 0$, which consequently adjusts the accumulated evidence $\sum_{t=0}^{T-1} \mathbf{e}_t$ such that the final Dirichlet expectation converges perfectly to the true posterior: $\frac{\alpha_{T,k}}{\sum_i \alpha_{T,i}} \to P(y=k \mid x)$.

C Difference Between NBSR and Classical Decision Trees

A fundamental distinction must be drawn between our NBSR framework and classical hierarchical models, such as Classification and Regression Trees (CART) (Breiman et al., 1984), Soft Decision Trees (SDTs) or hierarchical Mixtures of Experts (MoEs) (Jordan & Jacobs, 1993). The divergence lies in four critical architectural paradigms: the functional role of intermediate nodes, the mathematical viability of soft routing, the semantic nature of the input space, and the fundamental mechanism of information aggregation.

1. The Role of Intermediate Nodes: Routing Gates vs. Sequential Evidence Extractors In classical differentiable decision trees and hierarchical MoEs, the graph is composed of two strictly distinct types of nodes:

- **Internal Nodes (Routers):** these nodes do not produce any classification output. They act purely as routing gates that calculate transitional probabilities (e.g. 0.7 probability of routing left, 0.3 right).
- **Terminal Leaves (Experts):** these are the only nodes authorized to make a decision or output a class prediction.

In contrast, NBSR abandons this dichotomy. In our framework, each NBSR tree node, whether internal or terminal, consists of both a router and an evidence extractor (see Diagram 8). Consequently, intermediate nodes act as *both routers and experts*. When a sample arrives at an intermediate node (e.g. the Depth 1 `animal_mid` node in the CIFAR-10 classification task), the model does not merely calculate the probability of traversing to the `Pets` or `Wildlife` leaves. Instead, it immediately queries the global knowledge oracle $\mathbf{h}_x$ to extract a full K-dimensional evidence vector

(Eq.2):

$$\mathbf{e}_{\text{mid}} = \text{Activation}(\mathbf{W}_{\text{mid}}\mathbf{h}_x + \mathbf{b}_{\text{mid}})$$

This evidence is directly injected into the Dirichlet prior ($\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_0 + \mathbf{e}_{\text{mid}}$). By updating the belief state *at* the intermediate nodes before passing the computational flow deeper into the tree, the NBSR framework allows the decision to sequentially sharpen. This provides the mathematical foundation for dynamic early exiting, as the model can evaluate confidence at $t = 1$ without needing to reach a terminal leaf.

2. The Failure of Soft Routing in Sequential Frameworks The sequential nature of NBSR fundamentally breaks the standard "soft routing" paradigm used by classical neural trees.

In a classical hierarchical MoE, an image traverses every branch of the tree simultaneously. The network calculates a *Path Probability* for every terminal leaf by taking the product of the routing decisions along that branch. For example:

$$\pi_{\text{pets}} = P(\text{Animal} \mid \text{Root}) \times P(\text{Pets} \mid \text{Animal}) \tag{10}$$

The final prediction is then calculated by multiplying every terminal leaf's output by its respective path probability and summing them together to form a weighted continuous average:

$$\mathbf{y}_{\text{final}} = \sum_{\text{all leaves}} \pi_{\text{leaf}} \cdot \mathbf{y}_{\text{leaf}} \tag{11}$$

While this probability-weighted average of ensembles works for classical trees *where only leaves produce outputs*, it fails catastrophically in the NBSR framework. Because NBSR extracts evidence at *multiple depths*, applying standard soft routing would result in a severe entanglement of intermediate and leaf evidence:

$$\boldsymbol{\alpha}_T = \boldsymbol{\alpha}_0 + \sum_{\text{mid\_nodes}} \pi_{\text{mid}}\mathbf{e}_{\text{mid}} + \sum_{\text{leaf\_nodes}} \pi_{\text{leaf}}\mathbf{e}_{\text{leaf}} \tag{12}$$

If the router were allowed to output continuous probabilities (e.g. a 30%/30%/40% split), the final belief state would become a blended average of intermediate evidence layered on top of a blended average of leaf evidence. This compounding continuous relaxation inevitably leads to *representation collapse*. The local experts would fail to specialize, learning generic, overlapping features because the network relies on the blended ensemble to minimize the loss.

By enforcing **hard discrete routing** via the Gumbel-Softmax Straight-Through Estimator (STE), NBSR sidesteps this failure mode. The STE forces $\pi \in \{0, 1\}$ (i.e. 'hard routing'), ensuring that exactly one clean, sequential path of evidence (e.g. $\mathbf{e}_{\text{animal\_mid}} + \mathbf{e}_{\text{pets}}$) is extracted and added to the Dirichlet prior, thereby guaranteeing absolute expert specialization while retaining end-to-end differentiability.

3. The Input Space: Raw Partitioning vs. Semantic Oracle Querying In classical hierarchical models like CART (Breiman et al., 1984) and standard HMEs (Jordan & Jacobs, 1993; Waterhouse & Robinson, 1995), both the routing decisions and the expert predictions are typically computed directly from the raw input features x (e.g., via scalar thresholds or simple linear projections). The tree explicitly partitions this raw input space.

In contrast, NBSR introduces a *Persistent Global Knowledge Oracle* ($\mathbf{h}_x$). A shared deep neural backbone first maps the raw input into a dense, high-dimensional semantic space. This oracle is then broadcast to the entire graph - it's presented to all nodes at all depths. Consequently, the local experts in NBSR do not partition raw pixels or isolated tabular columns; instead, they act as *active attention mechanisms*, applying specialized learned filters ($\mathbf{W}_v$) to dynamically extract high-level conceptual evidence directly from the shared global representation at various stages of the reasoning process.

4. Aggregation Mechanism and Uncertainty Quantification: Multiplicative Probabilities vs. Additive Evidence In classical HMEs (Jordan & Jacobs, 1993; Waterhouse & Robinson, 1995), the final prediction is formed by *multiplying* gating probabilities along the path from root to leaf to form a path weight, which then weights the expert's probability distribution. Because this framework

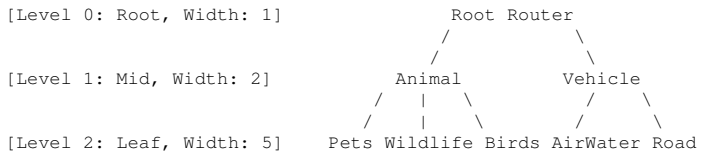
operates entirely within a zero-sum probability space ($\sum p = 1$), it forces the network to distribute a total probability mass of 1 across all outcomes, stripping the model of any native ability to quantify structural ignorance.

In contrast, NBSR operates in an *evidential* space. Instead of multiplying probabilities, NBSR *adds* continuous evidence vectors to a Dirichlet belief state ($\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t$). This additive accumulation breaks the zero-sum constraint during the intermediate steps, enabling the model to physically inflate the total volume of evidence (the Dirichlet precision, α_0). Consequently, NBSR natively provides rigorous Bayesian uncertainty quantification ($u = K/\alpha_0$). This allows the NBSR framework to explicitly express "I don't know" when encountering out-of-distribution data - a critical safety capability fundamentally absent in classical HMEs and traditional decision trees.

D Additional Experimental Results

D.1 CIFAR-10

We evaluate our sequential routing framework using a truncated ResNet-18 backbone configured under a *Predefined Taxonomy* graph structure. This human-logical hierarchy organizes the 10 CIFAR-10 classes by splitting them into semantic super-categories. At Depth 1, the graph routes between Animals (6 classes) and Vehicles (4 classes). At Depth 2, the graph allocates 5 specialized leaf experts to handle specific sub-populations: `Pets` (cat, dog), `Wildlife` (deer, horse, frog), `Birds` (bird), `AirWater` (airplane, ship), and `Road` (automobile, truck). The maximal graph topology is structured as follows:



Decoupling Graph Topology from the Outcome Space A common misconception regarding classical hierarchical decision trees is that the terminal leaf nodes (baskets) correspond directly to the final classes (e.g. 5 leaves equate to 5 classes), or that local experts output static, class-agnostic templates. In the NBSR framework, the graph topology is fully decoupled from the K-dimensional outcome space. Each leaf node represents an *Expert* (specialized neural network module), and every single expert, regardless of its position in the tree, computes a full 10-dimensional evidence vector. Thus each node, be it intermediate or terminal, is a decision-maker which produces the outcome (belief) with different levels of fidelity - they target being trained to be specialized consultants for one or more classes, but not all.

This means the number of leaf nodes (5) does not dictate the number of output classes (10). Instead, the 10 CIFAR-10 classes are distributed across the 5 leaf-node experts through the following learned specialization mapping:

- *Animals (6 Classes) → 3 Experts:* `Pets` (Cat, Dog), `Wildlife` (Deer, Horse, Frog), and `Birds` (Bird).
- *Vehicles (4 Classes) → 2 Experts:* `AirWater` (Airplane, Ship) and `Road` (Automobile, Truck).

Even though an expert like `Pets` specializes in a specific sub-population (cats and dogs), its output remains a full 10-dimensional vector. For example, consider an image of a dog routed down the predefined taxonomy. The root router identifies the input as an animal, and the subsequent router forwards it to the `Pets` expert. The `Pets` expert does not simply apply a static "+1 to Pets" update; rather, it acts as a dynamic function applied to the specific image's global features:

1. *The Oracle Holds Specifics:* the global knowledge oracle $\mathbf{h}_x$ encodes highly specific, high-dimensional visual concepts (e.g. floppy ears, a long snout, fur texture) rather than mere abstract categorical tags.

2. *Weights as Distinct Filters:* the expert calculates the evidence vector (Eq.2) $\mathbf{e}_{\text{pets}} = \text{Softplus}(\mathbf{W}_{\text{pets}}\mathbf{h}_x + \mathbf{b}_{\text{pets}})$. Importantly, $\mathbf{W}_{\text{pets}} \in \mathbb{R}^{10\times d}$ contains a distinct row vector for each of the 10 classes. The row vector $\mathbf{w}_{\text{dog}}$ learns to align precisely with dog-specific features, while $\mathbf{w}_{\text{cat}}$ aligns with cat-specific features.
3. *Asymmetric Evidence Extraction:* the inner product between $\mathbf{h}_x$ and $\mathbf{W}_{\text{pets}}$ yields a highly asymmetric 10-dimensional evidence vector. The dog dimension receives a massive positive scalar, the cat dimension receives a minor scalar (due to shared "furry" features), and evidence for unrelated classes (e.g. airplane and ship) is pushed to near-zero values.

Thus, the router's role is simply to delegate the computational flow to the expert equipped with the most highly-trained filters for that specific data sub-population. The chosen expert then independently interrogates the raw data to sharpen the Dirichlet distribution around a single specific class, preventing probability smearing across the broader sub-category.

Information Acquisition vs. Semantic Accuracy. The hyperparameter λ in the training objective Eq.3 governs a critical trade-off between the model's semantic accuracy and its rate of information acquisition. In practice, the Negative Log-Likelihood (NLL) remains the primary driver of feature discovery during early training. However, as the model converges and the NLL approaches its theoretical lower bound (asymptotic to $0.001 - 0.002$ by Epoch 100 in our tests), the entropy penalty $\mathcal{H}(\cdot)$ becomes the numerically dominant component of the loss function. This stage represents a distinct transition from *feature learning* to *belief sharpening*.

As illustrated in Fig. 2, the weighted structural pressure ($|\lambda \cdot \mathcal{H}|$) eventually overtakes the semantic loss and steadily climbs, crystallizing the model's confidence. In our experiments, a value of $\lambda = 10^{-3}$ provided a stable "Bayesian nudge", with which the NBSR model reached a peak accuracy of 96.74%.

To complement the quantitative results presented in the main text (Section.5.1), we provide the complete end-to-end training loss and test accuracy trajectories for the *Flat ResNet-18* (Fig. 9) and the *Sparse MoE* (Fig. 10) baselines over the 150-epoch training regime.

Notably, the Sparse MoE exhibits distinct variance and oscillation in its test accuracy, particularly in the later stages of training. This instability is a classic symptom of the continuous soft-routing gates competing against the classification experts, further highlighting the optimization friction introduced by standard MoE architectures on semantic classification tasks.

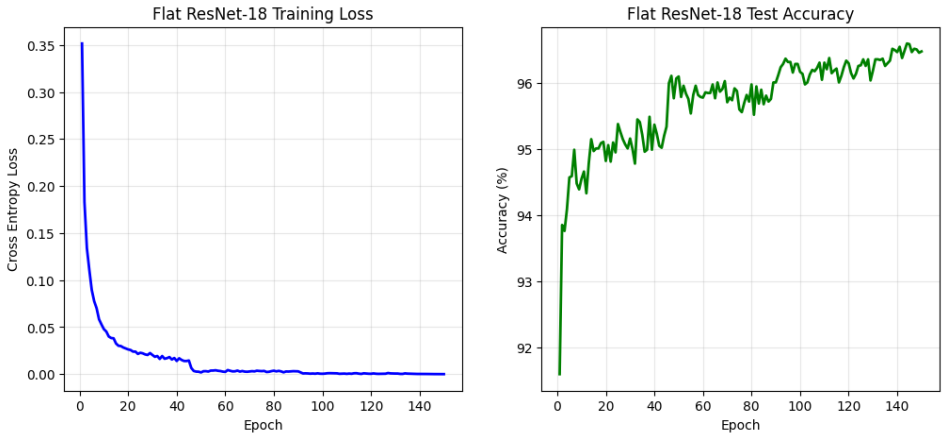


Figure 9: Training loss and test accuracy dynamics for the standard Flat ResNet-18 baseline on the CIFAR-10 dataset.

D.2 Structured Medical Diagnosis

Path-Dependent Feature Attribution. A standard critique of applying deep learning to tabular clinical data is the loss of feature-level interpretability. While baselines such as XGBoost natively

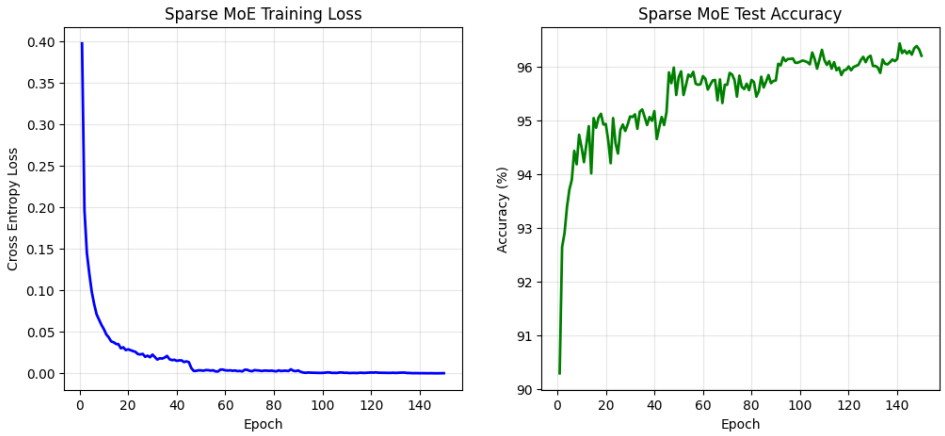


Figure 10: Training loss and test accuracy dynamics for the Sparse MoE (Soft Routing) baseline on the CIFAR-10 dataset. The noticeable jitter in the accuracy curve illustrates the optimization friction inherent in continuous soft-routing mechanisms.

provide feature importance via variance reduction, and general neural networks rely on post-hoc approximations (e.g. SHAP (Lundberg & Lee, 2017)). Our NBSR framework provides an exact, analytically differentiable feature attribution mechanism.

Because the terminal Dirichlet concentration for the predicted diagnosis y^* is a strict linear accumulation of the localized evidence vectors extracted along the active trajectory $\mathcal{T}$, defined as $\alpha_{T,y^*} = \alpha_{0,y^*} + \sum_{t=0}^{T-1} e_{t,y^*}$, we can calculate the exact feature importance vector $\mathcal{I}(x) \in \mathbb{R}^{|x|}$ using gradient-based attribution (i.e. input $\times$ gradient)[7]. By applying the chain rule through the global oracle $\mathbf{h}_x = f_\phi(x)$, the total feature importance perfectly decomposes into the specific contributions from each visited expert:

$$\mathcal{I}(x) = x \odot \frac{\partial \alpha_{T,y^*}}{\partial x} = x \odot \sum_{t=0}^{T-1} \left(\frac{\partial e_{t,y^*}}{\partial \mathbf{h}_x} \frac{\partial \mathbf{h}_x}{\partial x} \right) = x \odot \sum_{t=0}^{T-1} \left(\mathbf{J}_{\mathbf{h}_x}(x)^T \nabla_{\mathbf{h}_x} e_{t,y^*} \right) \tag{13}$$

where $\odot$ denotes element-wise multiplication, $\nabla_{\mathbf{h}_x} e_{t,y^*}$ is the gradient of the scalar evidence with respect to the oracle features, and $\mathbf{J}_{\mathbf{h}_x}(x)$ is the Jacobian matrix of the oracle representations with respect to the input.

This represents a fundamental shift from population-level statistics (e.g. XGBoost) to personalized medicine. As shown in Fig. 4, XGBoost outputs a *global* ranking of feature importance cumulated by the entire training set (e.g. highlighting 'congestion' or 'palpitations'). Conversely, NBSR dynamically generates an exact, *local* attribution specific to the input (i.e. the selected Patient 0), correctly identifying 'dischromic patches' and 'nodal skin eruptions' as the driving biometric rationales for their specific Fungal Infection diagnosis. This traces exactly *which sub-specialist expert* in the triage pathway utilized those specific features, yielding a fully transparent, step-by-step biomechanical rationale.

D.3 Language Modeling

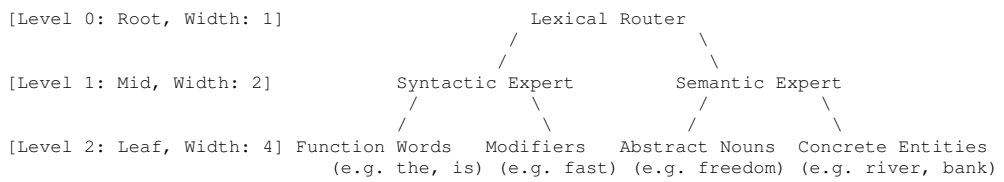
Modern Large Language Models (LLMs) generally operate via next-token prediction over a vocabulary space $\mathcal{V}$. However, standard monolithic LMs behave as opaque black boxes and frequently suffer from predictive overconfidence, leading to severe factual hallucinations (Ji et al., 2023; Guo et al., 2017; Desai & Durrett, 2020). While ad-hoc output mechanisms such as top-k (Fan et al.,

[7]In gradient-based attribution, the partial derivative quantifies the model's local *sensitivity* to a specific feature -a large gradient indicates that even minor perturbations in that feature's value would induce massive shifts in the predicted evidence. By multiplying this sensitivity by the feature's actual input magnitude (input $\times$ gradient), we obtain a first-order Taylor approximation of that feature's total additive contribution to the final diagnostic belief.

2018) or nucleus (top-p) (Holtzman et al., 2020) sampling[8] are commonly used to truncate the output distribution and inject diversity, they do not quantify the model's intrinsic epistemic uncertainty.

To evaluate NBSR's reasoning capacity and uncertainty estimation in sequence modeling, we design a contextual next-token prediction task. We utilize a controlled syntactic-semantic disambiguation corpus paired with a word-level tokenizer to maintain strict PoS boundaries. The text tokens are embedded and processed by a shared lightweight causal Transformer backbone to yield the contextual token representation $\mathbf{h}_x$.

Instead of a standard dense projection to the vocabulary logits, $\mathbf{h}_x$ is passed into the NBSR routing graph. The topology is designed to mirror human linguistic processing-first resolving broad part-of-speech (PoS) syntax (Jurafsky & Martin, 2000; Marcus et al., 1993), then refining via semantic context:



Interpretable Token Routing. A major advantage of NBSR in sequence modeling is the transparent resolution of linguistic ambiguity. In standard LMs, polysemous words or complex dependencies are resolved invisibly within dense attention matrices. NBSR, however, *physically routes the token prediction through specialized linguistic sub-spaces*.

To demonstrate this mechanism, we evaluate the models on an ambiguous but strictly **in-distribution** prompt context: *"the dog is by the..."*. The diagnostic output of the baseline models and our NBSR framework is captured in the following trace:

```
Prompt Context: "<bos> the dog is by the ..."

1. Standard Transformer -> Predicts: 'peace' (Black-box logit max)
2. Transformer MoE      -> Predicts: 'truth' (Discrete routing, no evidence trace)

3. NBSR Language Model  -> Predicts: 'war'
   [Phase 1] Root Router:
      -> Syntactic: 0.00 | Semantic: 1.00
   [Phase 2] Total Local Evidence Extracted by Leaves:
      -> Function: 0.0 | Modifier: 0.0 | Abstract: 275.5 | Concrete: 320.4
```

To a human reader, all three predictions lack semantic coherence. However, this perfectly aligns with the isolated nature of our controlled experiment. Because our synthetic corpus generates text via uniform random sampling across categorical grammar templates (e.g. `['function', 'concrete', 'function', 'function', 'abstract']`), semantic world-knowledge is deliberately absent, and predictions rely purely on learned syntactic structure. The models were trained on exactly 10,000 structurally valid but semantically nonsensical sentences, such as *"the car is by the logic"* or *"a tree was in the fear"*. The networks have absolutely zero real-world understanding of what a "dog" or a "river" is; they solely recognize the mathematical patterns of the grammar. Therefore, in this toy universe, predicting an abstract noun like *"war"* or *"peace"* is a 100% mathematically correct and valid syntactic completion.

The critical distinction lies in *interpretability*. While the standard Transformer blindly outputs a high logit for "peace" with zero explanation as to why, *NBSR provides a completely transparent, path-dependent reasoning trace*. The audit trail reveals that NBSR's Root router correctly deduced that a noun must follow the preposition, allocating 100% of its routing probability (1.00) to the Semantic Expert and successfully zeroing out the Syntactic Expert (i.e. probability 0.00). Subsequently, the leaf experts extracted massive diagnostic evidence specifically for Concrete (320.4) and Abstract (275.5) nouns, while explicitly extracting 0.0 evidence for grammatically invalid function words or modifiers.

[8] Top-k sampling restricts selection to the k most probable next tokens, whereas nucleus (top-p) sampling dynamically restricts selection to the smallest set of tokens whose cumulative probability exceeds a predefined threshold p. Both methods subsequently renormalize the truncated distribution before sampling.

Even though it ultimately outputs a token that sounds semantically anomalous to a human, the routing trace mathematically proves that NBSR successfully learned and executed the underlying grammatical rules. *It did not hallucinate an invalid part-of-speech* (like a verb); it predictably narrowed the universe down to a noun, *providing a step-by-step rationale that standard LLMs completely lack*. By utilizing this synthetic grammar, we perfectly isolated the interpretable routing mechanism, proving that NBSR physically routes tokens through a logical DAG (Part-of-Speech $\rightarrow$ Noun Type).

Uncertainty-Aware Generation *vs.* Top-k Selection. Traditional top-k sampling relies on softmax outputs, which are notoriously poorly calibrated (Guo et al., 2017; Desai & Durrett, 2020). Standard neural networks are essentially blind to the boundaries of their own knowledge. Because the softmax function mathematically forces all output probabilities to sum to 100%, the model is stripped of the ability to output "I don't know". When presented with an out-of-distribution (OOD) prompt, i.e. a scenario or token that fundamentally differs from the statistical universe the model was trained on, the network pushes the unfamiliar data through its layers and artificially inflates the highest random logit into a confident prediction (Hendrycks & Gimpel, 2017; Ovadia et al., 2019). This structural flaw is the mathematical root cause of AI hallucinations: a model confidently guessing on an unfamiliar input.

To evaluate this vulnerability, we feed the models an explicit OOD prompt containing an unknown token (<unk>):

```
Ambiguous OOD Prompt: "<bos> the <unk> ..."

1. Standard Transformer -> Predicts: 'or' with 6.7% confidence.
2. Transformer MoE      -> Predicts: 'to' with 6.6% confidence.
3. NBSR Language Model  -> Predicts: 'but' with 5.7% expected probability.
```

While our toy dataset's extremely limited vocabulary (65 tokens) naturally prevents the extreme 99% overconfidence spikes seen in massive LLMs, which causes all three models to output relatively flat distributions ($\sim$6%) for the unknown context, NBSR provides the *architectural* foundation to solve this scaling problem natively: it replaces the forced softmax with a Dirichlet distribution over the vocabulary space $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{V}|}$, which accumulates raw evidence rather than strictly forcing probabilities to sum to 1. When NBSR encounters an OOD prompt (like <unk>), it searches its specialized routing experts and finds zero matching evidence. Because it extracts no evidence, the total Dirichlet precision $\alpha_0 = \sum \alpha_i$ stays perfectly flat at the uniform prior.

As formalized in *Corollary 1 (Epistemic Collapse)*, this failure to accumulate evidence physically traps the system in a state of maximum epistemic uncertainty ($u \approx 1$). Therefore, *NBSR provides a native, mathematically rigorous measure of epistemic uncertainty*. This enables *Evidential Thresholding* as an *alternative* to top-k. Rather than blindly sampling from the top 50 tokens based on uncalibrated logits, the decoding algorithm can directly evaluate the Dirichlet precision α_0. When scaled to massive language corpora, if α_0 falls below a critical confidence threshold on an OOD prompt, the system mathematically recognizes that it lacks evidence. It can then gracefully halt, abstain, or ask the user for clarifying context, which *prevents hallucinations at the architectural level rather than relying on post-hoc output filtering*.

Efficacy of Early Exiting on Syntactic Tokens. Human language adheres to Zipf's law[9], meaning a vast majority of generated tokens are simple, highly predictable function words (e.g. *the, a, is, to*). Allocating the same amount of computational FLOPs to predict the word *"the"* as to predict a highly complex domain-specific noun is highly inefficient. *NBSR's dynamic early exiting acts as a resource-rational adaptive compute mechanism*. For highly predictable syntactic tokens, the Mid-Level *Syntactic Expert* extracts massive initial evidence, driving the Dirichlet entropy below the threshold η and triggering an early exit at Depth 1. Deep traversal (Depth 2) is automatically reserved only for complex, low-frequency semantic tokens requiring deeper contextual disambiguation. This dynamic depth allocation allows the fast configuration to accelerate token generation while maintaining competitive baseline perplexity.

[9]Zipf's law is an empirical principle stating that given a large corpus of data, the frequency of an item is inversely proportional to its rank ($f(r) \propto \frac{1}{r}$). Consequently, a small handful of highly ranked items (e.g. function words) dominate the vast majority of occurrences (Zipf, 1949).

D.4 NBSR-MEM

Standard deep reinforcement learning and imitation learning policies are notoriously brittle; they operate as opaque black boxes and suffer from catastrophic overconfidence when deployed in out-of-distribution (OOD) environments. Further, real-world autonomous agents rarely possess perfect global information. When an agent's observation space is restricted to a local window, the navigation task becomes a *Partially Observable Markov Decision Process* (POMDP) (Puterman, 1994). A purely *reactive* policy is vulnerable to *spatial amnesia* - once a crucial environmental cue (e.g. a wall or obstacle) leaves the local visual frame (i.e. the 'sensory' or 'receptive' field), the agent forgets its existence, leading to failure or infinite navigational loops. To resolve both the structural opacity of standard neural policies and the limitations of partial observability, we transition from *static* prediction to a dynamic control task and propose **NBSR-Mem**, which integrates a dynamic recurrent memory buffer into the evidential routing architecture.

D.4.1 TASK AND EXPERIMENTAL SETUP.

The autonomous agent is tasked with navigating a 2D grid environment to reach a randomized goal while avoiding walls and static obstacles. The environment is strictly partially observable: at any given timestep t, the agent receives only a 5×5 local spatial matrix centered on its current position (Fig.5). The action space consists of 4 discrete movements: 0 (Up/Cruising), 1 (Down/Reverse), 2 (Left/Evasion), and 3 (Right/Evasion). A purely reactive policy in this setting is vulnerable to spatial amnesia, i.e. once a crucial obstacle or landmark leaves the local visual frame, a reactive agent forgets its existence, which can lead to catastrophic failures or infinite navigational loops.

To isolate and test the memory mechanism under partial observability, we utilize a procedurally generated sequence of abstracted keyframes representing the crucial phases of a corridor traversal. A single trajectory is structured as follows (as illustrated in Fig. 5):

- $t = 0$ (Memory Cue): the agent receives an initial visual cue (a sign on the left or right wall) that dictates whether it must turn left or right at a distant intersection.
- $t = 1, 2$ (Spatial Amnesia Zone): the agent receives a featureless spatial matrix, representing transit through an empty corridor. A purely reactive agent loses all historical context during this phase.
- $t = 3$ (Intersection): the agent arrives abruptly at an intersection with a wall directly ahead. To successfully navigate, it must recall the cue from $t = 0$ and turn in the correct direction.

Training Paradigm: Behavioral Cloning. We employ Behavioral Cloning (BC) (Pomerleau, 1988), a foundational method of Imitation Learning (Hussein et al., 2017), to train the agent, as illustrated in Fig. 11. BC reframes *dynamic control* as a *supervised classification* task, where the goal is to learn a policy π_θ that mimics the actions of an algorithmic expert (which possesses perfect global information). The algorithmic expert generates a dataset of optimal state-action pairs (o_t, a_t^*). The neural actor then learns to map state observations directly to the expert's discrete actions. During training, the agent policy, π_θ, samples a batch of observations o_t and outputs a predicted action $\hat{a}_t$. The network parameters θ are updated via *Backpropagation Through Time* (BPTT) to minimize the Negative Log-Likelihood (NLL) between the predicted action distribution and the expert's true action a_t^*.

We deliberately selected this training paradigm over standard Reinforcement Learning (RL) for two critical reasons:

- *Architecture Isolation*: by removing the high variance, reward-shaping dependencies, and hyperparameter sensitivity inherent to RL, any improvements in performance or safety can be strictly attributed to the NBSR topology.
- *Covariate Shift Testbed*: standard BC policies are notoriously susceptible to covariate shift - they mimic the expert blindly and fail catastrophically when encountering novel OOD states. This makes BC an ideal testbed for demonstrating NBSR's native evidential abstention mechanisms.

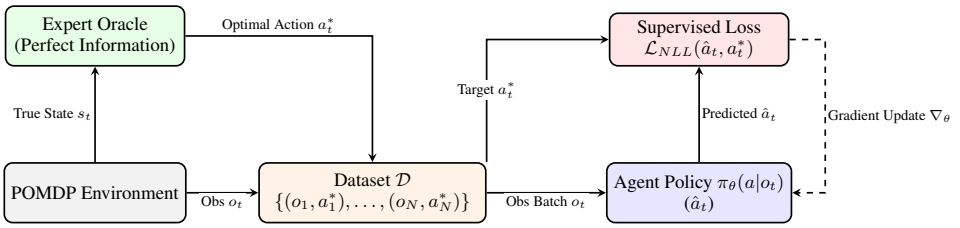


Figure 11: The Imitation Learning (Behavioral Cloning) paradigm under partial observability. An expert oracle utilizes the true state (s_t) to generate optimal actions (a_t^*), which are paired with the agent's limited observations (o_t) to form the dataset. The agent policy π_θ is trained offline using supervised learning to mimic the expert, isolating architectural efficacy from Reinforcement Learning instabilities.

NBSR-Mem and Baselines. To construct **NBSR-Mem**, we couple the *Global Knowledge Oracle* (a lightweight CNN) with a *Gated Recurrent Unit* (GRU). Importantly, the memory update occurs *prior* to expert routing. The CNN features update a persistent hidden state $\mathbf{m}_t$ encoding the historical trajectory. This *temporally-aware state* $\mathbf{m}_t$ is then passed onto the hierarchical evidential DAG (illustrated below), where expert networks are intentionally flattened to single linear layers to ensure unhindered gradient flow.

```
[Level 0: Root, Width: 1]                    Navigational Router
                                              /                 \
                                             /                   \
[Level 1: Mid, Width: 2]             Cruising Expert          Evasion Expert
                                      /         \              /          \
                                     /           \            /            \
[Level 2: Leaf, Width: 4]         Up           Down        Left          Right
                             (Straight)     (Reverse)    (Avoid)       (Avoid)
```

To validate the necessity of both temporal memory and evidential routing, we benchmark NBSR-Mem against 3 distinct architectural configurations:

1. *Standard CNN (Reactive):* a standard feed-forward convolutional network utilizing a Softmax output. Lacking memory, it is expected to succumb to spatial amnesia; lacking evidential routing, it acts as an opaque black-box prone to OOD overconfidence.
2. *CNN-GRU (Memory):* a temporal baseline where the CNN features update a GRU hidden state before passing to a standard linear classifier. While capable of solving the POMDP task via memory, it remains structurally opaque and highly vulnerable to OOD hazards.
3. *NBSR (Reactive):* the hierarchical evidential routing architecture *without* the GRU memory buffer. It is expected to fail the primary navigation task due to spatial amnesia, but successfully trigger epistemic safety halts when encountering OOD traps.

Training Dynamics: Stabilizing the Recurrent Routing Training this *Recurrent Evidential Routing Network* is historically challenging (Shazeer et al., 2017; Pascanu et al., 2013). Naive implementations typically suffer from *expert collapse* (Shazeer et al., 2017), vanishing gradients (Hochreiter & Schmidhuber, 1997; Pascanu et al., 2013), and temporal memory amnesia (catastrophic forgetting) (Bengio et al., 1994). To stabilize NBSR-Mem, we implement 4 critical structural interventions:

1. *Deterministic Routing for BPTT:* we replace the stochastic Gumbel-Softmax with a deterministic, standard Softmax exclusively during training. This creates a mathematically smooth pathway, allowing *Backpropagation Through Time* (BPTT) to flow seamlessly from the routing leaves, through the hierarchy, and deep into the GRU to link past states to future decisions.
2. *Semantic Concept Routing (Auxiliary Loss):* without explicit guidance, hierarchical routers, particularly when utilizing a deterministic Softmax, seek the laziest optimization,

routing all states to a single, monolithic "master expert" (*Expert Collapse*) [10]. To prevent this, we apply a lightweight *Auxiliary Concept Loss* ($\mathcal{L}_{route}$) that physically forces the Root Router to map to human-interpretable sub-spaces: straight corridors *must* be routed to the "Cruising" expert, while complex intersections *must* trigger "Evasion."

3. *Masked Evidential Regularization:* standard evidential regularizers penalize all generated evidence, creating a "Gradient Tug-of-War" where the NLL loss encourages correct evidence, but the regularizer punishes it, resulting in memory failure. We utilize a target-masked regularizer (Sensoy et al., 2018) that only penalizes hallucinated evidence on *incorrect* classes. This forces resource rationality, squashing incorrect actions strictly to zero, while leaving the correct class free to soar to massive confidence. This shows that the original entropy regularization term $\lambda \sum_{v_t \in \mathcal{T}} \mathcal{H}(\boldsymbol{\alpha}_t)$ in Eq.3 can sometimes introduce optimization deadlock and requires masking for calibration.
4. *Targeted Weight Decay and LayerNorm:* to anchor the latent space and bound out-of-distribution explosions, we apply LayerNorm and targeted L2 weight decay to the CNN backbone. This ensures untrained visual channels (e.g. alien hazards - see later) mathematically decay to zero, preventing random noise from being amplified by the network.

Interpretable Policy Routing *vs.* Black-Box Control. When the standard CNN-GRU policy navigates an intersection, it outputs an action with *no mechanistic explanation*. In contrast, NBSR-Mem physicalizes the decision-making process. By examining the audit trace of a successful evasive maneuver, we observe the model explicitly routing through semantic sub-spaces: (The trace is shown in Section 5.4.) Thanks to the *Auxiliary Concept Loss*, Phase 1 perfectly isolates the abstract mode (Evasion: 1.00). Further, because the model is resource-rational, the evidence is sharp and sparse - it outputs exactly 0.0 for incorrect actions, proving that the evidential regularizer successfully stopped all hallucinations.

Resource-Rational Adaptiveness and Computational Efficiency. Notably, the integration of temporal memory profoundly impacts dynamic routing depth. The reactive NBSR policy, lacking historical context at intersections, exhibits high epistemic uncertainty. Because a perfectly uncertain 4-class Dirichlet distribution ($\boldsymbol{\alpha} = [1, 1, 1, 1]$) yields a maximum differential entropy of roughly -1.79, the reactive policy safely exceeds the -4.5 confidence threshold we set, forcing the network to route to Depth 2 (Avg. Depth: 1.18)[11].

In contrast, NBSR-Mem utilizes its GRU hidden state to achieve extreme confidence. As seen in the trace above, the Depth 1 Evasion Expert alone extracts a massive 28.3 evidence for the correct turn based on memory. This profound reduction in uncertainty causes the entropy to plummet past the threshold, allowing the router to safely early-exit. Remarkably, the model achieves sufficient epistemic confidence to early-exit on all states, yielding an average inference depth of exactly 1.00 while maintaining 100% accuracy. Thus, our architecture demonstrates that resolving partial observability not only recovers task performance but actively reduces the computational footprint of the hierarchical policy.

Epistemic Safety in Out-of-Distribution Environments. The most critical failure mode of standard autonomous policies is their behavior in novel environments. To test this, we inject an "Alien Hazard" (a visual channel never present in training) directly into the agent's path.

Because standard baselines utilize a softmax activation, the mathematical constraint ($\sum p = 1$) forces the policy to confidently hallucinate. In 100% of our OOD trials, the standard CNN-GRU policy selected a movement action and crashed into the hazard (Table 4).

[10] While deterministic Softmax exacerbates mode collapse due to its "winner-take-all" gradient scaling, expert collapse is fundamentally an optimization pathology where it is mathematically cheaper to update an already-competent expert than to train a novel one. This "rich get richer" dynamic occurs across routing mechanisms (including Sigmoid, Gumbel-Softmax, and Top-K) unless explicit load-balancing or semantic guidance is applied.

[11] The average depth is calculated across the entire test trajectory dataset. Because the reactive policy lacks memory, it successfully early-exits on the trivial empty corridors (comprising the majority of states), but is forced to traverse to Depth 2 upon reaching every intersection due to high epistemic uncertainty. This blend of Depth 1 and Depth 2 routing yields an overall episodic average of 1.18.

NBSR-Mem natively solves this through Dirichlet thresholding, empirically validating *Corollary 1 (Epistemic Collapse)*. Due to our *LayerNorm* and *targeted weight decay*, the untrained Alien channel weights were mathematically anchored to zero. When the Alien Trap appeared, the network extracted only 8.39 total precision - safely below the operating threshold of 10.0 we set. Because the evidence remained stagnant, the epistemic uncertainty u mathematically refused to collapse. Instead of hallucinating, NBSR-Mem recognized its own ignorance and triggered a safe fallback protocol (100.0% Halt). Without these target-masked regularizations, ID and OOD evidence distributions dangerously overlap; with them, NBSR-Mem perfectly preserves the epistemic safety margin. Ultimately, this demonstrates that NBSR-Mem "knows when it doesn't know", providing a robust, uncertainty-aware framework for safe autonomous control.

D.5 AR-NBSR

The NBSR framework natively operates as a sequential BOED engine. At any inference step t, the routing action a_t formally corresponds to selecting the next experimental design ξ (e.g. querying a specific sub-specialist or running a distinct medical test), and the extracted evidence $\mathbf{e}_t$ serves as the observation. Historically, maximizing EIG in deep neural networks is computationally intractable because the posterior entropy requires evaluating an intractable marginal likelihood, often necessitating complex variational lower bounds (Foster et al., 2019). However, because the NBSR Bayesian state update is defined via exact conjugate addition ($\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t$), our framework bypasses this intractability entirely. The exact, realized information gain achieved by querying expert v_{t+1} is analytically computed as the drop in differential entropy: $\Delta\mathcal{H}_t = \mathcal{H}(\boldsymbol{\alpha}_t) - \mathcal{H}(\boldsymbol{\alpha}_{t+1})$.

As established in Proposition 1, the structural penalty $\lambda \sum \mathcal{H}(\boldsymbol{\alpha}_t)$ in our training objective intrinsically forces the router to maximize this sequential EIG by penalizing delayed uncertainty reduction. By introducing explicit, asymmetric measurement costs into the graph, we can extend NBSR into a fully resource-rational *Active Learning* framework.

To empirically validate NBSR as an active BOED agent, we extend the Structured Medical Diagnosis task (Section 5.2) into an *Active Clinical Triage* environment. In standard machine learning, the model is given free, simultaneous access to all input features. In this experiment, the patient's true diagnostic state is initially hidden. The agent only has access to free, baseline demographic data at $t = 0$.

The 132 symptoms are clustered into 5 distinct "Diagnostic Test Panels" (e.g. Basic Metabolic Panel, Neurological Exam, Dermatological Swab). Each panel is governed by a dedicated NBSR Expert node and has an associated financial or temporal cost $c(v)$, detailed in Table 6.

Table 6: Diagnostic Test Panels and Associated Costs ($c(v)$). These values represent the asymmetric financial or temporal constraints required to execute each specific diagnostic panel.

Diagnostic Panel	**Assigned Cost** $c(v)$
Basic Metabolic Panel (Blood)	10.0
Hepatic Panel	15.0
Respiratory Swab	20.0
Neurological Exam	30.0
MRI Scan	50.0

Connection to the Standard EIG Objective. To understand why optimizing Eq. 5 effectively solves the active learning problem, we compare it directly to the standard BOED objective. Recall that standard BOED seeks a design ξ that maximizes the Expected Information Gain (Foster et al., 2019):

$$\text{EIG}(\xi) = I(y; e \mid \xi) = \mathbb{E}_{p(e|\xi)}\big[\mathcal{H}(p(y)) - \mathcal{H}(p(y \mid e, \xi))\big]$$

In the NBSR framework, at step t, the "prior" state is given by the Dirichlet distribution parameterized by $\boldsymbol{\alpha}_t$. The routing decision a_t serves as the experimental design ξ, and the extracted evidence $\mathbf{e}_t$ serves as the observation e. The exact, realized information gain of taking step t is the reduction in entropy: $\Delta\mathcal{H}_t = \mathcal{H}(\boldsymbol{\alpha}_t) - \mathcal{H}(\boldsymbol{\alpha}_{t+1})$.

Minimizing the cumulative sum of entropies $\lambda \sum_{t=0}^{T} \mathcal{H}(\boldsymbol{\alpha}_t)$ in Eq. 5 forces the network to minimize the area under the entropy curve. Algebraically, this heavily penalizes delayed uncertainty reduction, forcing the routing policy π_θ to consistently select paths that maximize the expected stepwise drop $\mathbb{E}[\Delta\mathcal{H}_t]$ at every juncture. Therefore, our regularized loss $\mathcal{L}_{\text{BOED}}$ acts as a computationally tractable, end-to-end surrogate for maximizing the exact sequential EIG (the EIG objective). This circumvents the intractable marginal likelihood integrations typically required in standard variational BOED (Foster et al., 2019).

Autoregressive Routing (AR-NBSR) to Prevent Combinatorial Explosion. Deploying a static, feed-forward DAG topology into this active triage environment would induce a massive combinatorial explosion. Specifically, given N available diagnostic test panels (the branching factor) and a maximum required testing sequence of L steps (the tree depth), a hardcoded decision tree would require an intractable $\mathcal{O}(N^L)$ leaf-node complexity. To resolve this, we configure the framework as an *Autoregressive State Machine (ASM)*.

We formally define the AR-NBSR as an ASM - a sequential generative system where the current state $\mathcal{S}_t$ and the subsequent transition are conditioned on the history of prior observations. We define the ASM by the tuple $(\mathcal{S}, \mathcal{A}, \pi_\theta, \mathcal{U})$, where $\mathcal{S}_t = (\mathbf{h}_x, \boldsymbol{\alpha}_t, \mathbf{m}_t)$ is the joint embedding of oracle features, current Dirichlet belief state, and action history; $\mathcal{A}$ is the action space of diagnostic panels; $\pi_\theta(a_t \mid \mathcal{S}_t)$ is the routing policy; and $\mathcal{U}$ is the state-update function defined by the Bayesian conjugate addition $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t$. While the state evolution inherently satisfies the Markov property (as the posterior belief $\boldsymbol{\alpha}_{t+1}$ is conditioned solely on the previous state $\boldsymbol{\alpha}_t$ and the instantaneous evidence $\mathbf{e}_t$), we term this process *autoregressive* because the model uses its own past evidential accumulations to regress towards a terminal hypothesis.

Conceptually, this maps perfectly to the cyclical architecture illustrated in Fig. 6: instead of routing a patient down a physical hallway of isolated sub-specialists (a deep DAG), a single reusable *Global Triage Router* and a flat array of independent Expert Networks operate in a recurrent, time-indexed loop. Consequently, static architectural *depth* is replaced by recurrent *time*. The diagnostic trajectory unfolds dynamically as an iterative loop, for example:

1. $t = 0$**:** The Global Router evaluates the initial baseline prior $\boldsymbol{\alpha}_0$ alongside the demographic features, and selects the first test (e.g. Panel 2). Panel 2 extracts evidence $\mathbf{e}_0$, updating the belief to $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_0 + \mathbf{e}_0$.

2. $t = 1$**:** The exact same Global Router evaluates the newly updated state $\boldsymbol{\alpha}_1$. Recognizing lingering uncertainty regarding a specific pathology, it selects a complementary test (e.g. Panel 4). Panel 4 extracts evidence $\mathbf{e}_1$, updating the belief to $\boldsymbol{\alpha}_2 = \boldsymbol{\alpha}_1 + \mathbf{e}_1$.

3. $t = 2$**:** The router evaluates $\boldsymbol{\alpha}_2$. The newly accumulated evidence causes the differential entropy $\mathcal{H}(\boldsymbol{\alpha}_2)$ to plunge below the confidence threshold η. The recurrent loop terminates dynamically, yielding the final diagnosis without consuming further testing budget.

This autoregressive formulation elegantly bounds the parameter count to a single router and N experts, while granting the agent a dynamic, virtually infinite sequential action space.

Clinical Metaphor: The Active Triage Room

To intuitively map the mathematical terminologies of AR-NBSR to a real-world clinical setting, consider a hospital triage room:

- **The Global Router (The Lead Diagnostician):** Rather than passing the patient down a physical hallway of isolated sub-specialists (a deep DAG), a single Lead Diagnostician sits at a central desk. Their only job is to evaluate the current information and decide *which test to order next*.
- **The Expert Networks (The Test Panels):** Down the hall are 5 distinct testing rooms (e.g. Blood Panel, MRI, Respiratory Swab). These are the expert modules.
- **The Dirichlet Belief State $\boldsymbol{\alpha}_t$ (The Clipboard):** The Diagnostician holds a clipboard tracking the probability of 41 possible diseases. It starts blank (maximum entropy).
- **Autoregressive Time (The Loop):** The Diagnostician sends the patient to Room 2. The result (evidence $\mathbf{e}_0$) comes back, and the clipboard is updated ($\boldsymbol{\alpha}_1$). Realizing more information is needed to distinguish between two highly probable diseases, the Diagnostician sends the patient to Room 4 ($\mathbf{e}_1 \rightarrow \boldsymbol{\alpha}_2$).
- **The Confidence Threshold η (The Stop Condition):** Once the clipboard indicates sufficient certainty for a specific disease, the Diagnostician halts the testing loop, sparing the patient the financial cost and time of visiting the remaining 3 rooms.

Baselines. We benchmark the NBSR active learning agent against two standard sequential acquisition strategies:

1. **Random Allocation:** Iteratively queries random test panels until the entropy threshold η is reached.
2. **Greedy EIG (Myopic BOED):** A standard active learning baseline that evaluates the expected entropy drop for all available tests at step t, and selects the test that maximizes the immediate gain-to-cost ratio $\frac{\Delta \mathcal{H}_t}{c(v)}$. While optimal for a single isolated step, myopic BOED is well-known to lack long-horizon planning.

Training Dynamics and Convergence. The training stability of the AR-NBSR model is evidenced by the convergence curves in Fig. 7. We observe a precipitous decline in NLL during the initial 40 epochs, indicating that the expert networks successfully internalized the diagnostic features of the noisy clinical dataset. Concurrently, the *Average Patient Cost* descends from the maximum budgetary ceiling ($125) to a stable operational regime ($\sim$ $48). This confirms that the AR-NBSR agent reaches a stable, resource-rational equilibrium, balancing predictive accuracy against the cost-awareness regularizer without exhibiting signs of training instability or divergence.

Dynamic Uncertainty Reduction. A hallmark of Bayesian Optimal Experimental Design is that the optimal measurement strategy is inextricably dependent on the prior. Our audit trail (Fig. 7) provides a *non-myopic proof* of this behavior. The fact that entropy drops significantly across multiple steps ($t = 0$ to $t = 5$) confirms the model is not relying on a "magic bullet" test but rather gathering a chain of evidence, where each test updates the belief state, and the subsequent test selection is conditioned on the evidence gathered previously. The threshold crossing at $t = 2$ demonstrates the model performing dynamic early-exiting: it recognized it had attained sufficient information to be confident and halted testing to save resources.

Conclusion on Empirical Performance. As demonstrated in our cost-accuracy Pareto Frontier, AR-NBSR exhibits strong resource-rationality. While the Greedy baseline provides a strong myopic upper bound, AR-NBSR achieves competitive diagnostic accuracy with significantly lower average computation per inference. Further, the audit trail confirms that AR-NBSR does not merely execute a fixed sequence, but dynamically reduces belief entropy across multiple time steps, naturally terminating upon reaching the confidence threshold η. This confirms that our framework is capable of long-horizon planning, moving beyond the myopic limitations of standard active learning.

E Extended Discussion

E.1 Active Knowledge Retrieval vs. Information Bottlenecking

Standard deep neural networks frequently suffer from information bottlenecking (Shwartz-Ziv & Tishby, 2017). When data is presented entirely at the input layer, critical low-level details can be prematurely discarded as information propagates through sequential transformations. For example, early layers in a convolutional neural network may extract broad spatial features while irreparably discarding granular textures that might be vital for downstream, fine-grained classification. While residual connections partially mitigate this by passing signals across contiguous layers, they do not preserve the unadulterated global context across the entire depth of the network.

In contrast, the NBSR framework fundamentally bypasses this degradation by treating the initial dense embedding as a *Persistent Knowledge Oracle* ($\mathbf{h}_x$). Instead of processing the data in a single monolithic pass where information is passively lost, the decision graph actively and iteratively queries this oracle. In this sense, NBSR shares conceptual similarities with Joint-Embedding Predictive Architectures (JEPA) (LeCun, 2022): both systems seek to filter environmental noise and extract abstract, task-relevant true signals. However, in NBSR, this extraction is performed sequentially and information retrieval is done repeatedly. Because each nodal decision-maker selectively chooses exactly what knowledge to retrieve from the global oracle to resolve its immediate uncertainty, this process inherently functions as an *attention mechanism* over the semantic space. Every local nodal expert $f(\mathbf{h}_x; \mathbf{W}_{v_{t+1}}, \mathbf{b}_{v_{t+1}})$ serves as a task-specific probe, dynamically projecting the static, high-dimensional representation into a localized, outcome-discriminative evidence space.

E.2 Hierarchical Epistemic Capacity and Safe Inference

To accurately simulate a hierarchical taxonomy, different levels of abstract detail must be absorbed at different depths. As empirically demonstrated in our Toy Experiment (Section 5), this is mathematically enforced by manipulating the *epistemic capacity* of the local experts. By applying a scaled *Sigmoid* activation to intermediate nodes, we impose a strict "confidence budget", forcing mid-level experts to act as generally broad, hesitant super-categories. Conversely, terminal leaf experts utilize an unbounded *Softplus* activation, allowing them to inject massive evidence to crystallize a sharply defined decision boundary.

A critical architectural imperative in this deployment is the strict prohibition against normalizing these intermediate Dirichlet evidence vectors ($\boldsymbol{\alpha}_t$). In standard neural control policies, latent normalization (e.g. via a continuous Softmax) is routinely applied to bound representations. However, in Subjective Logic, the total Dirichlet precision α_0 serves as the fundamental mathematical anchor for epistemic uncertainty, defined as $u = K/\alpha_0$. Artificially normalizing $\boldsymbol{\alpha}_t$ forces α_0 to a constant, permanently destroying the framework's native ability to trigger out-of-distribution (OOD) safety abstentions. By ensuring that evidence remains strictly raw, additive, and unbounded, NBSR successfully avoids hallucination vulnerabilities, guaranteeing safe epistemic collapse (Corollary 1) as validated in our POMDP Control and Language Modeling experiments.

E.3 NBSR as an Markov Decision Process (MDP)

While NBSR is heavily evaluated on discriminative classification tasks, we can formally characterize its sequential inference process as a discrete-time Markov Decision Process (MDP) (Puterman, 1994) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. Here, the state space $\mathcal{S}$ is defined by the augmented belief state $\boldsymbol{\alpha}_t$ and the persistent oracle context $\mathbf{h}_x$; the action space $\mathcal{A}$ corresponds to the discrete selection of expert nodes via the router $\pi_\theta(a_t \mid \mathcal{S}_t)$; the transition dynamics $\mathcal{P}(\mathcal{S}_{t+1} \mid \mathcal{S}_t, a_t)$ are governed by the exact Bayesian conjugate update $\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t + \mathbf{e}_t$; and the reward function $\mathcal{R}$ is implicitly defined as the negative differential entropy $-\mathcal{H}(\boldsymbol{\alpha}_t)$ plus any associated measurement costs $c(v_t)$.

In this MDP, the optimal policy π^* seeks to maximize the cumulative information gain while minimizing expenditure, effectively treating diagnostic triage and sequential reasoning as an optimal control problem. This MDP formulation reveals that our NBSR architecture is fundamentally a *Policy Network* learning to navigate an epistemic state-space, providing the algorithmic foundation for the Autoregressive State Machine (AR-NBSR) demonstrated in our Active Clinical Triage experiment (Section 5.5).

E.4 Modular Skill Acquisition and Unbounded Topologies

As the field of artificial intelligence shifts towards modular, agentic systems, architectures must evolve to support composable skill ensembles. In NBSR, each expert network acts as an isolated, highly specialized cognitive skill. The routing tree ensembles these skills both horizontally and sequentially, selectively activating only the specific sub-regions required to process the current input. This mechanism offers a computationally lightweight, Bayesian alternative to standard dense attention mechanisms.

Looking forward, this framework naturally accommodates several rigorous theoretical extensions:

- **Neuro-Symbolic Integration via Product of Experts:** the sequential evidence accumulation natively supports the injection of externally derived rules or heuristics. Future work could calibrate the node Dirichlet distribution via a *Product of Experts*[12] (PoE) (Hinton, 2002; Huang, 2026), using a distinct correction term derived from the contextual side information ($\boldsymbol{\varepsilon}_t$).
- **Unbounded Width via Dirichlet Process:** Instead of routing over a fixed K-dimensional categorical distribution, routers could be reformulated using a *Dirichlet Process* via a differentiable *stick-breaking mechanism*[13] (Ishwaran & James, 2001). In this paradigm, a router leaves a residual probability mass to instantiate a newly discovered expert, allowing the horizontal branching factor to grow infinitely and organically if existing experts fail to sufficiently minimize the negative log-likelihood.
- **Unbounded Depth via Information-Gain Splitting:** To dynamically expand vertical depth, we can apply node-splitting rules from traditional decision trees (e.g. CART (Breiman et al., 1984)) to our neural modules. If a terminal expert's average differential entropy $\mathcal{H}(\boldsymbol{\alpha}_T)$ plateaus above a critical threshold (which indicates it lacks the capacity to resolve the remaining uncertainty of its assigned data sub-population), the network can dynamically freeze that expert, spawn a new router in its place, and branch into newly initialized child experts to further partition the semantic space.

F Experimental Details

F.1 CIFAR-10

Computing Environment. All experiments, training regimes, and inference benchmarking were conducted on a high-performance Linux server (x86_64 architecture) equipped with a 24-core (48 logical threads) AMD EPYC 9B45 processor and 190 GB of system memory. GPU acceleration was provided by a single NVIDIA RTX PRO 6000 Blackwell Server Edition with 96 GB of VRAM, running CUDA version 13.0 and NVIDIA Driver 580.82.07. To ensure strict reproducibility across all baselines, identical deterministic seeds (Seed = 111) were applied to Python's 'random', 'NumPy', and 'PyTorch' libraries, alongside the enforcement of deterministic CuDNN backend algorithms.

Network Architectures. The *visual feature extraction backbone* for all models is an unfrozen ResNet-18 (He et al., 2016), initialized with ImageNet weights. The network is truncated prior to its final classification head, yielding a 512-dimensional global feature vector $\mathbf{h}_x$ for each input image.

In our Neural Bayesian Sequential Routing (NBSR) framework, the routers and experts are instantiated as follows:

- *Routers:* each internal node acts as a gating mechanism parameterized by a *Multi-Layer Perceptron* (MLP). The router concatenates the visual features $\mathbf{h}_x$ with the current Dirich-

[12] PoE is a machine learning framework that models a probability distribution by combining the output from several simpler probability models (experts), this is achieved by multiplying their probability distributions and renormalizing.

[13] Stick-breaking processes are widely used procedures for constructing random discrete distributions in statistics and machine learning (Gil-Leyva et al., 2026). Due to their intuitive construction and computational tractability, they have become popular in modern Bayesian nonparametric inference, serving as the foundational mechanism for models such as the Dirichlet and Pitman-Yor processes.

let concentration state $\boldsymbol{\alpha}_t$, passing them through a 128-unit hidden layer with a *ReLU activation*, followed by a *linear projection* to the number of branching paths. The output is sampled via a continuous *Gumbel-Softmax approximation* during training.

- *Local Experts:* each expert module (e.g. `animal_mid`, `pets`, `wildlife`) consists of a single *linear transformation* ($\mathbb{R}^{512\to 10}$) followed by a *Softplus activation* function. The Softplus nonlinearity is strictly required to ensure that the extracted evidence vector remains positive ($e_k > 0$), preserving the mathematical integrity of the Dirichlet parameter updates.

Hyperparameters and Optimization. All models were trained for a total of 150 epochs with a batch size of 128. We optimized the networks using the *Adam optimizer* (Kingma & Ba, 2017) with an initial learning rate of 2×10^{-4}, decayed by a factor of $\gamma = 0.5$ every 45 epochs via a StepLR scheduler. Gradients were clipped at a maximum norm of 1.0 to ensure stability.

For the data pipeline, the CIFAR-10 images were upscaled to 224×224 to match the expected resolution of the ImageNet-pretrained ResNet backbone. Standard spatial augmentations were applied during training, including *random horizontal flipping*, *random rotation* ($\pm 15^\circ$), and *color jittering* (brightness $= 0.2$), followed by standard *channel-wise normalization*.

For the NBSR specific hyperparameters, the structural entropy penalty weight was set to $\lambda = 10^{-3}$. To balance early-stage structural exploration with late-stage discrete routing, the Gumbel-Softmax temperature τ was annealed exponentially per epoch from an initial value of 1.0 down to a minimum threshold of 0.1 at a decay rate of 0.97.

F.2 Evaluation metric

We report the Expected Calibration Error (ECE) (Guo et al., 2017) with $M = 10$ equal-width confidence bins. For the deterministic baselines the confidence is the maximum softmax probability; for NBSR it is the largest Dirichlet expected marginal $\max_k \alpha_{T,k}/\alpha_0^{(T)}$.

F.3 Structured medical diagnosis

Here we provide the technical details, network architectures, and hyperparameters required to reproduce the structured tabular clinical experiments presented in Section.5.2.

F.3.1 Computing Environment

All experiments, including baseline training and NBSR evaluation, were executed on a cloud-based virtual machine instance (*Google Colab*) to measure CPU-bound inference efficiency. The hardware specifications are as follows:

- *Processor:* Intel(R) Xeon(R) CPU @ 2.20GHz (1 Physical Core, 2 Logical Threads)
- *System Memory (RAM):* 13.61 GB
- *Hardware Accelerator:* none (CPU-only execution)
- *Software Stack:* Python 3.12, PyTorch 2.10.0, and XGBoost 2.0.

F.3.2 Dataset and Preprocessing

We utilized the Kaggle Disease Symptom Prediction dataset[14], which maps patient profiles to specific clinical endpoints.

- *Input Features (X):* 132 binary indicators representing the presence (1) or absence (0) of specific clinical symptoms (e.g. 'skin_rash', 'joint_pain').
- *Target Classes (Y):* 41 discrete disease categories.

[14] `https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning`

- *Clinical Noise Injection:* real-world Electronic Health Records (EHR) are inherently noisy due to entry errors, missing tests, and ambiguous patient reporting. To simulate this environment and prevent the models from overfitting to a perfectly deterministic toy dataset, we applied a uniform 5% noise mask to both the training and testing sets. Specifically, for 5% of the feature matrix X, the binary symptom states were inverted ($X_{i,j} \leftarrow 1 - X_{i,j}$).

F.3.3 NETWORK ARCHITECTURES

Flat MLP Baseline. The standard monolithic deep learning baseline was constructed using a sequential stack of dense layers:

1. Linear($132 \to 128$), ReLU activation
2. Linear($128 \to 128$), ReLU activation
3. Linear($128 \to 41$), Softmax activation

NBSR Architecture. The NBSR framework utilized a maximum depth of 2 (Root $\to$ Mid $\to$ Leaf) with a uniform branching factor of 4. The modular components were parameterized as follows:

- *Global Oracle Backbone:* identical to the MLP hidden layers to ensure fair semantic capacity: Linear($132 \to 128$) $\to$ ReLU $\to$ Linear($128 \to 128$) $\to$ ReLU.
- *Router Networks:* MLP processing the concatenated feature state $\mathbf{h}_x$ and current belief $\boldsymbol{\alpha}_t$: Linear($169 \to 64$) $\to$ ReLU $\to$ Linear($64 \to 4$), followed by a Gumbel-Softmax activation.
- *Expert Networks (Evidence Extractors):* linear($128 \to 41$) followed by a Softplus activation to ensure strictly non-negative evidence vectors $\mathbf{e}_t > 0$.

F.3.4 OPTIMIZATION AND HYPERPARAMETERS

Both the Flat MLP and NBSR networks were trained using *Adam optimizer* (Kingma & Ba, 2017) with a learning rate of 1×10^{-3} and a batch size of 128 for 40 epochs. A global random seed of 111 was enforced across data splits, noise injection, and network initializations.

For the XGBoost baseline, we utilized the default 'XGBClassifier' parameters with 'eval_metric='mlogloss'' and 'use_label_encoder=False'.

Specific NBSR hyperparameters were set as follows:

- *Routing Temperature (τ):* the Gumbel-Softmax temperature was exponentially annealed per epoch to smoothly transition from soft exploration to hard, discrete routing: $\tau = \max(0.1, 1.0 \cdot 0.9^{\text{epoch}})$.
- *Entropy Penalty (λ):* to prevent artificial overconfidence on the heavily perturbed noisy clinical data, the explicit entropy minimization penalty was disabled ($\lambda = 0.0$). The network relied purely on the Negative Log-Likelihood (NLL) of the expected Dirichlet probability.
- *Early Exiting Threshold (η):* during inference, the "Deep" configuration bypassed early exiting (threshold evaluated as 'None'), forcing the full architectural depth. The "Fast" configuration utilized a differential entropy threshold of $\eta = -100.0$ to dynamically truncate routing on unambiguous clinical presentations.

F.4 LANGUAGE MODELING

Here we provide the technical details, network architectures, and hyperparameters required to reproduce the synthetic language modeling and contextual disambiguation experiments presented in Section.5.3.

F.4.1 COMPUTING ENVIRONMENT

All sequence modeling experiments were deliberately executed on a cloud-based virtual machine instance (Google Colab). The hardware specifications are as follows:

- *Processor:* Intel(R) Xeon(R) CPU @ 2.20GHz (1 Physical Core, 2 Logical Threads)
- *System Memory (RAM):* 13.61 GB
- *Hardware Accelerator:* none (CPU-only execution)
- *Software Stack:* Python 3.12, PyTorch 2.1.0

F.4.2 Dataset and Preprocessing

To perfectly isolate syntactic reasoning from semantic world-knowledge, we constructed a synthetic contextual disambiguation corpus.

- *Vocabulary ($|\mathcal{V}|$):* 65 discrete tokens, strictly partitioned into 4 special tokens (`<pad>`, `<bos>`, `<eos>`, `<unk>`), 19 function words, 14 modifiers, 14 abstract nouns, and 14 concrete entities.
- *Generative Templates:* sequences were procedurally generated using uniform random sampling across 5 predefined syntactic templates (e.g. `['function', 'concrete', 'function', 'function', 'abstract']`).
- *Data Splits:* The dataset was split into 10,000 training sequences and 2,000 held-out test sequences. All sequences were padded to a maximum length of 6 tokens.

F.4.3 Network Architectures

Shared Causal Transformer Backbone. All three evaluated models (Standard, MoE, NBSR) share an identical Transformer backbone to ensure fair semantic capacity. It consists of an embedding layer ($d_{model} = 64$) with positional encoding, followed by a 2-layer causal Transformer Encoder utilizing 4 attention heads and a feed-forward dimension of 256.

Standard Transformer Baseline. The monolithic baseline processes the backbone's contextual representation $\mathbf{h}_x$ through a standard dense linear projection: Linear($64 \to 65$), followed by a standard Log-Softmax activation.

Transformer MoE Baseline. The MoE baseline utilizes a discrete Top-1 routing head without evidence accumulation. It consists of a router network—Linear($64 \to 4$) with Softmax, and 4 independent expert networks parameterized as Linear($64 \to 65$).

NBSR Architecture. The NBSR framework utilizes a hierarchical Directed Acyclic Graph (DAG) with a maximum depth of 2. The modular components were parameterized as follows:

- *Root Router:* an MLP processing the concatenated state $\mathbf{h}_x$ and the uniform Dirichlet prior $\boldsymbol{\alpha}_0$: Linear($129 \to 32$) $\to$ ReLU $\to$ Linear($32 \to 2$), followed by a Gumbel-Softmax activation.
- *Mid-Level Experts (Syntax vs. Semantics):* 2 distinct experts, each structured as Linear($64 \to 64$) $\to$ ReLU $\to$ Linear($64 \to 65$), followed by a *Softplus* activation to ensure strictly non-negative evidence vectors $\mathbf{e}_t > 0$.
- *Leaf Experts (PoS Categories):* 4 terminal experts mapping to the distinct linguistic subspaces (Function, Modifier, Abstract, Concrete). These are identical in structure to the mid-level experts.

F.4.4 Optimization and Hyperparameters

All networks were trained using the *Adam optimizer* (Kingma & Ba, 2017) with a learning rate of 1×10^{-3} and a batch size of 64 for 5 epochs. A global random seed of 42 was enforced across data generation and network initializations.

Specific NBSR hyperparameters were set as follows:

- *Routing Temperature (τ):* the Gumbel-Softmax temperature was exponentially annealed per epoch to smoothly transition from soft exploration to hard, discrete routing: $\tau = \max(0.1, 1.0 \cdot 0.9^{\text{epoch}})$.

- *Early Exiting Threshold (η):* during inference, the "Deep" configuration bypassed early exiting (threshold evaluated as 'None'), forcing the full architectural depth. The "Fast" configuration utilized a differential entropy threshold of $\eta = 1.0$ to dynamically halt routing on predictable function words.
- *OOD Abstention Threshold (τ_{conf}):* the critical confidence threshold for evaluating the epistemic uncertainty of the total Dirichlet precision on out-of-distribution prompts was defined mathematically as $\tau_{conf} = 1.5 \times |\mathcal{V}| = 97.5$.

F.5 POMDP NAVIGATION

Here we provide detailed specifications for the experimental environment, dataset generation, model architectures, and training hyperparameters utilized in the sequential control and planning task (Section 5.4).

F.5.1 COMPUTING ENVIRONMENT

All experiments, training, and evaluations were conducted on a Google Colab virtual instance. To ensure reproducibility and demonstrate the lightweight nature of the NBSR-Mem architecture, the experiments were executed entirely on the CPU without hardware acceleration. The environment specifications are as follows:

- *CPU:* AMD EPYC 7B12 (1 physical core, 2 logical threads), 2.25 GHz base clock.
- *Memory:* 13.61 GB RAM.
- *Software Stack:* Python 3.12.13, PyTorch 2.10.0+cpu.

F.5.2 POMDP DATASET GENERATION

The navigation task was framed as a classification problem via Behavioral Cloning. We procedurally generated a synthetic dataset of optimal state-action sequences representing a Partially Observable Markov Decision Process (POMDP).

State Representation: each state observation o_t is a $4 \times 5 \times 5$ spatial tensor representing a local grid centered on the agent. The 4 channels encode distinct environmental features (e.g. floor, agent position, left-turn cues, right-turn cues).

Trajectory Structure: each sequence has a fixed length of $T = 4$ timesteps, structured to enforce reliance on temporal memory:

1. $t = 0$ *(Memory Cue):* the agent receives a cue dictating the ultimate turning direction. A random variable determines if the sequence is "simple" (agent moves straight indefinitely) or "complex" (agent must evade a wall at the end). For complex sequences, a target direction (Left or Right) is selected, and a corresponding visual cue is instantiated in the spatial tensor.
2. $t = 1, 2$ *(Spatial Amnesia Zone):* the agent traverses a featureless corridor. The optimal action is 0 (Cruising/Straight). The visual cue from $t = 0$ is no longer present in the observation.
3. $t = 3$ *(Intersection):* if the sequence is complex, a wall appears directly ahead of the agent. The optimal action is the target direction designated at $t = 0$. If the sequence is simple, the corridor remains empty, and the optimal action remains 0.

Out-of-Distribution (OOD) Generation: To test epistemic safety, an "Alien Hazard" was generated by injecting a high-magnitude activation into a specific coordinate of the fourth visual channel - a feature entirely absent from the training distribution.

Dataset Splits: The generated dataset comprised 5,000 training sequences and 1,000 test sequences. The probability of a sequence being "complex" (requiring a turn) was set to 70% to heavily penalize purely reactive models.

F.5.3 Model Architectures

All models share a common *Global Knowledge Oracle* (CNN Backbone) to ensure a fair comparison of routing and memory capabilities.

CNN Backbone: The feature extractor consists of two convolutional layers followed by a linear projection:

- `Conv2d`: 4 input channels, 32 output channels, kernel size 3×3, padding 1.
- `Conv2d`: 32 input channels, 64 output channels, kernel size 3×3, padding 0.
- `Linear`: 576 ($64 \times 3 \times 3$) to 128 dimensions.
- `LayerNorm`: applied to the 128-dimensional output without elementwise affine parameters to strictly anchor the latent space.
- *Activations:* ReLU is applied after all convolutional and linear layers. *Note: Biases were removed from all layers to improve evidential calibration.*

Action Space and Network Outputs: the theoretical action space $\mathcal{A}$ is defined by four discrete movements: 0 (Cruising/Straight), 1 (Reverse), 2 (Evasion Left), and 3 (Evasion Right). To enable gradient-based optimization, the network architectures output continuous 4-dimensional vectors representing their predictive confidence. For the standard baselines (CNN and CNN-GRU), the final classifier outputs a vector of log-probabilities via a Log-Softmax activation. Conversely, the NBSR architectures output a vector representing the concentration parameters of a Dirichlet distribution, $\boldsymbol{\alpha} = \mathbf{e} + \mathbf{1}$, where $\mathbf{e} \geq 0$ denotes the accumulated evidence for each action. This distinction is critical for safety: while a highly uncertain baseline is mathematically forced to distribute probabilities that sum to 1 (e.g. $[0.25, 0.25, 0.25, 0.25]$), the NBSR model can output $\boldsymbol{\alpha} = [1.0, 1.0, 1.0, 1.0]$, explicitly quantifying a state of total epistemic uncertainty (zero evidence) to trigger a safe halt. During training, the discrete expert action a_t^* is transformed into a one-hot vector to properly isolate the penalty applied by the masked evidential regularizer.

Memory Module (CNN-GRU & NBSR-Mem): for models equipped with temporal memory, the 128-dimensional output of the CNN backbone is processed sequentially by a standard Gated Recurrent Unit (`nn.GRU`) with an input size of 128, a hidden size of 128, and 'batch_first=True'.

Hierarchical Evidential DAG (NBSR & NBSR-Mem): the routing topology consists of three levels:

- *Root Router:* a single `Linear(128, 2)` layer mapping the hidden state to the two abstract modes (Cruising vs. Evasion).
- *Mid-Level Experts:* two `Linear(128, 4)` layers (without biases), one for each abstract mode. These extract initial Dirichlet evidence ($\mathbf{e}_{mid}$).
- *Leaf Experts:* four `Linear(128, 4)` layers (without biases). These extract additive Dirichlet evidence ($\mathbf{e}_{leaf}$) if the Depth 1 entropy remains above the threshold.

During training, the Root Router utilizes a standard Softmax activation. During inference, it uses an argmax (one-hot) to enable discrete early exits. All experts utilize a Softplus activation to ensure non-negative evidence generation.

F.5.4 Training Protocol and Hyperparameters

All models were trained via Behavioral Cloning (Pomerleau, 1988) using the Adam optimizer (Kingma & Ba, 2017). To stabilize the Recurrent Evidential Routing Network, we employed a specific combination of losses and regularizers.

Hyperparameters:

- *Batch Size:* 64
- *Epochs:* 30

- *Learning Rate:* 2×10^{-3}
- *Weight Decay:* 2×10^{-3} applied *exclusively* to the CNN Backbone parameters.
- *Base Dirichlet Prior ($\boldsymbol{\alpha}_0$):* $\mathbf{1}$ (a tensor of ones).
- *Inference Entropy Threshold (η):* -4.5 (used for full dataset evaluation to enforce deep traversal on intersections; -0.5 for the single OOD trace).
- *OOD Abstention Threshold (τ_{conf}):* 10.0

Loss Formulation (NBSR Models): the training objective is a composite of three terms:

$$\mathcal{L}_{total} = \mathcal{L}_{NLL} + \mathcal{L}_{reg} + \lambda_{route}\mathcal{L}_{route} \tag{14}$$

1. *Negative Log-Likelihood ($\mathcal{L}_{NLL}$):* computed using the expected probability of the final Dirichlet distribution: $\mathbf{p} = \boldsymbol{\alpha}_T / \sum \boldsymbol{\alpha}_T$.
2. **Auxiliary Concept Loss ($\mathcal{L}_{route}$):** a standard Cross-Entropy loss applied to the Root Router logits to enforce semantic mapping. We map the optimal action $a_t^* \in \{0, 1, 2, 3\}$ to an abstract target mode $m_t^* \in \{0, 1\}$ corresponding to the hierarchical routing tree. Specifically, $m_t^* = 0$ if $a_t^* \in \{0, 1\}$ (Cruising/Straight/Reverse) and $m_t^* = 1$ if $a_t^* \in \{2, 3\}$ (Evasion/Left/Right). The loss is defined as:

$$\mathcal{L}_{route} = -\sum_{j=0}^{1} \mathbb{I}(m_t^* = j) \log(p_{root,j}) \tag{15}$$

   where $p_{root,j}$ is the predicted probability of mode j from the Root Router. The weighting factor was set to $\lambda_{route} = 0.1$.
3. *Masked Evidential Regularization ($\mathcal{L}_{reg}$):* to penalize hallucinated evidence without causing gradient starvation, we utilized the target-masked regularizer (Sensoy et al., 2018):

$$\mathcal{L}_{reg} = \lambda_{reg}^{(e)} \frac{1}{K} \sum_{i=1}^{K} (\alpha_{T,i} - 1)(1 - y_i) \tag{16}$$

   where $\mathbf{y}$ is the one-hot encoded target action. To allow the GRU time to form initial memory pathways, we applied Evidential Annealing, linearly ramping the regularizer weight $\lambda_{reg}^{(e)}$ from 0.0 to 0.02 over the first 10 epochs.

F.6 BOED ACTIVE CLINICAL TRIAGE

Here we provide the technical specifications and hyperparameter configurations required to reproduce the Active Clinical Triage experiments presented in Section 5.5.

F.6.1 COMPUTING ENVIRONMENT

All experiments were executed on a cloud-based virtual machine instance (KVM full virtualization) to specifically isolate and measure CPU-bound inference efficiency. The exact hardware specifications are as follows:

- *Processor:* Intel(R) Xeon(R) CPU @ 2.20GHz, x86_64 architecture (1 Physical Core, 2 Logical Threads) with 55 MiB L3 cache.
- *System Memory (RAM):* 13.61 GB
- *Storage:* 242.49 GB allocated disk space
- *Hardware Accelerator:* None (Strict CPU-only execution; no GPU utilized)

F.6.2 DATASET AND PREPROCESSING

We utilized the Kaggle Disease Symptom Prediction dataset (same as used in Section 5.2), mapping clinical symptom profiles to 41 distinct pathologies.

- *Input Features (X):* 132 binary indicators representing symptom presence (1) or absence (0).
- *Partitioning:* Features were partitioned into 12 base demographic features (Always visible) and 5 Diagnostic Test Panels, each containing 24 task-specific symptom indicators.
- *Clinical Noise:* A uniform 5% noise mask was applied to the feature matrix, where binary states were stochastically inverted ($0 \leftrightarrow 1$).

F.6.3 Network Architectures

AR-NBSR Architecture.

- *Global Router:* An MLP processing the concatenated state $[\mathbf{h}_x, \boldsymbol{\alpha}_t, \mathbf{m}_t]$, where $\mathbf{m}_t$ is the one-hot history of previously queried panels. Architecture: Linear($178 \to 128$) $\to$ LayerNorm $\to$ ReLU $\to$ Linear($128 \to 5$).
- *Expert Networks (Panels):* Independent feature extractors: Linear($36 \to 128$) $\to$ LayerNorm $\to$ ReLU $\to$ Linear($128 \to 128$) $\to$ ReLU $\to$ Linear($128 \to 41$) $\to$ Softplus.

F.6.4 Optimization and Training Dynamics

Networks were optimized using Adam (Kingma & Ba, 2017) (lr $= 10^{-3}$) for 200 epochs with a batch size of 128. A *ReduceLROnPlateau* scheduler (factor $= 0.5$, patience $= 10$) was employed to stabilize convergence.

Hyperparameters.

- *Entropy Penalty (λ):* 1×10^{-4}.
- *Budget Penalty (γ):* 0.005.
- *Gumbel-Softmax Temperature (τ):* Exponentially annealed from 1.0 to 0.1.
- *Training Early Exiting (η):* -150.0.
- *Evaluation Sweep (η):* For the Pareto Frontier, we evaluated a dense grid of 30 linearly spaced confidence thresholds $\eta \in [-110.0, -400.0]$.