Variational Neurons and Bayesian Neural Networks: Distinct Probabilistic Objects, Inference Locations, and Computational Granularity

Yves Ruffenach[0009−0009−4737−0555]

Conservatoire National des Arts et Métiers, Strasbourg, France
yves@ruffenach.net

Abstract. Probabilistic neural architectures can reuse priors, Kullback–Leibler (KL) terms, reparameterized samples, Monte Carlo evaluation, and predictive distributions while assigning them to different random objects and computational locations. This article introduces an operational framework that separates four non-equivalent axes—Bayesianity, variationality, stochasticity, and distribution-valued computation—and uses six questions to identify the random object, inference or regularization location, conditioning information, sampling location, propagated representation, and prediction rule. It formalizes Definition 1, a generic unit-level specification criterion for a *variational neuron*: a hidden computational unit containing an input-conditioned local variational latent variable, a declared prior or reference distribution, a locally scoped forward map, a separately computable local cost, and diagnostics at the same declared unit index. Clause-level applications to published third-party architectures illustrate how the criterion distinguishes example-, representation-, weight-, output-, and unit-level probabilistic loci. EVE (*Elemental Variational Expanse*) provides the controlled worked implementation. Its mechanistic experiments show fixed-checkpoint functional sensitivity, latent-allocation changes with dimensionality, and covariance tracking under controlled relocation of observation noise. A locked fresh 50-seed, exact 2,472-parameter comparison with a heteroscedastic output model yields paired MSE, NLL, CRPS, and MACE intervals including zero, showing closely matched predictive performance at the attained sensitivity. Together, the framework and worked evidence provide a traceable method for separating structural specification, functional use, causal relevance, semantic sensitivity, and predictive utility in probabilistic neural architectures.

Keywords: Bayesian neural networks · variational inference · variational neurons · local latent variables · EVE · computational granularity · uncertainty quantification

1 Introduction

Neural networks can be made probabilistic in several fundamentally different ways. A model may place a distribution over network parameters, introduce latent variables at the data, layer, sequence, or unit level, randomize hidden activations, predict an output distribution, average independently trained networks, or combine several of these mechanisms. These constructions can all produce samples or uncertainty scores, yet they need not describe the same unknown quantity or perform the same form of inference. The terms *Bayesian*, *variational*, *stochastic*, *probabilistic*, and *distributional* should therefore not be treated as interchangeable [1,17].

Shared devices—priors, KL terms, reparameterization, and Monte Carlo sampling—do not imply shared probabilistic semantics. A standard BNN represents dataset-conditioned uncertainty over parameters or functions and predicts by marginalizing those unknowns; a variational neuron instead places an input-conditioned local latent state inside hidden computation. The decisive distinction is the probabilistic object and its computational location.

This distinction does not make the two constructions mutually exclusive. A local variational state may be embedded in a broader Bayesian latent-variable model, and a network may combine uncertainty over weights with local latent variability. The correct label follows from the explicitly declared random objects, conditioning evidence, inferential targets, and prediction rule.

This article makes three contributions: (i) a four-axis taxonomy organized by six reporting questions; (ii) a generic unit-level specification criterion for variational neurons that separates structural placement of a local variational latent variable from posterior claims; and (iii) a controlled empirical case study, instantiated with EVE (Elemental Variational Expanse), testing fixed-checkpoint functional use, latent dimensionality, sensitivity to manipulated noise structure, paired predictive effects, and exact-capacity comparison with an output-heteroscedastic model.

The primary contribution is an operational reporting framework for distinguishing probabilistic objects, inference loci, sampling loci, propagated representations, and prediction rules. Definition 1 specializes this framework to unit level by giving an operational criterion for using the architectural term *variational neuron* precisely, distinct from layer-, representation-, parameter-, and output-level formulations. EVE serves as the concrete implementation used for empirical instantiation, while the generic architectural term remains *variational neuron*. The case study focuses on fixed-checkpoint use, latent-dimensional allocation, response to a controlled change in the data-generating process, and predictive differences after exact parameter matching within one deliberately controlled regression family. Cross-architecture applicability is demonstrated structurally in Section 5 and through a clause-level application of Definition 1 to published third-party architectures in Appendix A. Sections 4.6 and Appendix C further distinguish the implemented variationally regularized regime from Bayesian extensions developed for taxonomic completeness.

1.1 Relation to earlier variational-neuron studies

The first EVE study introduced the local variational computational unit with an input-conditioned latent distribution, reference regularization, reparameterized local computation, and locally observable variational quantities [41]. The second studied the design space created by latent dimensionality, local capacity control, and temporal persistence [42]. The present article adds a distinct layer of formalization: the four-axis/six-question cross-architecture reporting framework, Definition 1 and its partition rule, the posterior-claim rule, and the staged A–E claim–evidence hierarchy.

This organizing perspective complements UQ surveys centered on uncertainty sources and semantics, method families, measures, calibration, and benchmark practice [17,20]. The architecture-reporting protocol jointly records random object, inference location, conditioning, sampling site, propagated representation, and prediction rule, together with a predeclared unit-local routing boundary and progressively stronger evidence claims. The controlled EVE study supplies worked evidence for those levels, while the cross-architecture tables and clause-level third-party application exercise the structural criterion outside the EVE design itself.

2 Four Distinct Axes of Probabilistic Neural Computation

The taxonomy is multi-label: each architecture is evaluated independently along the four axes before any global label is assigned. Section 5 applies this framework to representative probabilistic architectures.

Each construction is described through six fields:

1. Which object is random?
2. Where is inference or regularization located?
3. On which evidence is the distribution conditioned?
4. Where does sampling occur?
5. What representation is propagated to the next computation?
6. How is the final prediction formed?

Operational use. To classify a new architecture, first fill these six fields; then identify the random object and inference/regularization locus, locate sampling and the propagated object, and assign the four axis labels. For a unit-level claim, subsequently apply Definition 1 clause by clause. Ambiguous cases are assigned to the coarsest computational granularity directly supported by the declared architecture. The clause-by-clause record makes each decision auditable; Table 6 applies the procedure to four published third-party architectures spanning example-, representation-, weight-, and output-level variational computation.

A candidate axis is retained only when architectures can differ on that property without a necessary corresponding change on the others. Bayesian updating, variational optimization, sampled computation, and distribution-valued representation therefore remain distinct.

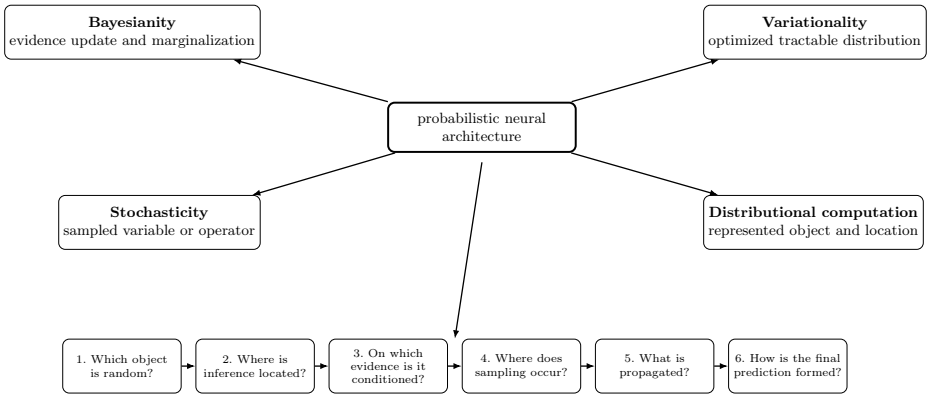

Fig. 1. Classification map for probabilistic neural architectures. The four axes are non-equivalent descriptive dimensions. The six questions identify the random object, inference location, conditioning evidence, sampling location, propagated representation, and prediction rule.

2.1 Bayesianity

Bayesian modelling specifies uncertain quantities, assigns priors, conditions on observations, and forms posterior or posterior-predictive quantities by integration. In neural networks, the uncertain quantities may be weights, functions, hyperparameters, local latent variables, or combinations thereof [5,29,33,49]. Bayesianity is therefore a property of the probabilistic model and its inferential semantics, not of one particular neural architecture.

2.2 Variationality

Variational inference (VI) replaces an intractable inference problem by optimization over a tractable family [10]. It can approximate a posterior over weights, a posterior over latent variables, a filtering distribution, or another target density. Conversely, Bayesian inference can be performed by Markov chain Monte Carlo (MCMC), Laplace approximation, expectation propagation, sequential Monte Carlo, or other methods without VI [19,24,40,47]. Variationality is therefore not synonymous with Bayesianity.

2.3 Stochasticity

A stochastic computation samples a random variable or uses a randomized operator. Dropout masks, noisy activations, latent codes, randomized gates, and posterior weight samples all satisfy this broad criterion. Stochasticity alone does not identify the source, inferential meaning, or uncertainty semantics of the randomness [7,23,30,48].

Table 1. Four complementary concepts in probabilistic neural computation.

Concept	Defining question	Typical mathematical object	Properties requiring separate identification
Bayesianity	Which unknowns receive priors, are conditioned on data, and are marginalized?	$p(\theta \mid \mathcal{D})$, $p(z \mid x, \mathcal{D})$, posterior-predictive integrals	Approximation algorithm, distributional family, and neural implementation
Variationality	Which tractable distribution is optimized relative to a target density?	$q_\lambda(\theta)$ or $q_\phi(z \mid x)$ and an evidence lower bound (ELBO)	Bayesian semantics and interpretation of the target density
Stochasticity	Which random variable or randomized operator is sampled?	Noise ϵ, masks, gates, weights, or latent states	Posterior semantics, calibration, and uncertainty interpretation
Distributional computation	At what architectural location is a distribution represented and used?	A local family $q_i(z_i \mid h_i)$, a sample, or distributional statistics	Random object, conditioning, propagated representation, and prediction rule

2.4 Distributional computation and granularity

A deterministic neuron maps an input representation to a point-valued activation. A distribution-valued hidden unit parameterizes or samples from a local distribution that participates in the forward computation. This architectural property does not by itself determine whether the distribution represents epistemic uncertainty, aleatoric uncertainty, ambiguity, controlled variability, compression, or another learned latent structure.

The declared random object, sampling site, propagated representation, and decision output are distinct descriptive roles:

$$\text{random object} \not\equiv \text{sampling location} \not\equiv \text{propagated object} \not\equiv \text{decision output}. \tag{1}$$

For example, the local reparameterization trick samples a pre-activation even though the modeled random object remains a weight posterior. Conversely, a local latent distribution may be sampled inside a unit and reduced to a point-valued activation before the next layer. Sampling location therefore does not determine inferential identity.

The non-implications are immediate. A model can be Bayesian but not variational, as in an MCMC-trained BNN; variational but not a BNN, as in a VAE with point-estimated weights; stochastic but neither Bayesian nor variational, as in fixed noise injection; or distributional only at the output, as in heteroscedastic regression. Hybrid architectures may combine all four properties.

3 Bayesian Neural Networks: Posterior Uncertainty over Models

3.1 Canonical formulation

Let $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ be a dataset and let w denote network parameters. A canonical BNN specifies a prior $p(w)$ and likelihood $p(\mathcal{D} \mid w) = \prod_n p(y_n \mid x_n, w)$.

Bayes' rule gives

$$p(w \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid w)p(w)}{p(\mathcal{D})}, \qquad p(\mathcal{D}) = \int p(\mathcal{D} \mid w)p(w)\, dw. \tag{2}$$

For a new input x_*,

$$p(y_* \mid x_*, \mathcal{D}) = \int p(y_* \mid x_*, w)\, p(w \mid \mathcal{D})\, dw. \tag{3}$$

The defining operation is posterior marginalization over plausible models or functions, not merely the existence of a distribution somewhere in the network [33,49].

Exact inference is generally intractable. Bayes by Backprop learns a variational distribution over weights [11,18]; probabilistic backpropagation uses assumed-density and expectation-propagation ideas [19]; stochastic gradient Langevin dynamics (SGLD) approximates posterior sampling [47]; Laplace methods approximate the posterior around a mode [40]; and Stochastic Weight Averaging-Gaussian (SWAG) fits a Gaussian approximation from an optimization trajectory [31]. These methods differ, but their inferential object remains model parameters or induced functions conditioned on the dataset.

3.2 Weight-space and observation uncertainty

Weight uncertainty and input-dependent observation variance are distinct probabilistic objects. Their standard decomposition and function-space context are retained in Appendix C; the main text proceeds to the unit-level construction because the paper's contribution concerns computational location rather than a new BNN uncertainty decomposition.

4 Variational Neurons: Unit-Level Variational Latent Computation

4.1 Variational inference and amortized latent variables

For observed variables x and unknowns u with joint density $p(x, u)$, VI selects a tractable family $\mathcal{Q}$ and solves

$$q^*(u) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(u) \,\|\, p(u \mid x))\,. \tag{4}$$

Equivalently, it maximizes

$$\mathcal{L}(q) = \mathbb{E}_{q(u)}[\log p(x, u)] - \mathbb{E}_{q(u)}[\log q(u)] = \log p(x) - \mathrm{KL}(q(u) \| p(u \mid x)). \tag{5}$$

The optimization machinery is agnostic to whether u denotes weights, a global code, a trajectory, or a local hidden state [9,10].

A VAE specifies $p_\theta(x, z) = p(z)p_\theta(x \mid z)$ and an amortized family $q_\phi(z \mid x)$ [27,39]:

$$\mathcal{L}_{\mathrm{VAE}}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \| p(z))\,. \tag{6}$$

For a diagonal Gaussian,

$$q_\phi(z \mid x) = \mathcal{N}\big(\mu_\phi(x), \mathrm{Diag}(\sigma_\phi^2(x))\big), \qquad z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{7}$$

Importance weighting and normalizing flows provide richer variational families [12,38]. A conventional VAE usually places the latent at the example or model-component level. The same variational machinery can instead be attached explicitly to a declared hidden computational unit. In this article, *variational neuron* is the generic architectural term for such a unit when it satisfies the unit-level specification criterion below; the term alone does not assert Bayesian posterior semantics.

A KL term alone does not establish Bayesian posterior inference. For

$$\mathcal{J} = \mathcal{L}_{\text{task}} + \beta \,\mathrm{KL}(q_\phi(z \mid h) \| p(z)), \tag{8}$$

the expression is a negative ELBO when the task term is the negative expected log-likelihood of a specified joint model, $\beta = 1$, and q_ϕ approximates the corresponding posterior. Otherwise, it is more accurately described as variational regularization.

4.2 Local latent computation

Let $h_i \in \mathbb{R}^{d_i}$ be the deterministic input to hidden unit i. A minimal variational neuron introduces $z_i \in \mathbb{R}^{k_i}$ and

$$q_{\phi_i}(z_i \mid h_i, c_i) = \mathcal{N}\big(\mu_{\phi_i}(h_i, c_i), \mathrm{Diag}(\sigma_{\phi_i}^2(h_i, c_i))\big), \tag{9}$$

where c_i may contain contextual, layer-level, or temporal information. A sample is obtained by

$$z_i = \mu_{\phi_i}(h_i, c_i) + \sigma_{\phi_i}(h_i, c_i) \odot \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I), \tag{10}$$

and participates explicitly in

$$a_i = F_{\eta_i}(h_i, z_i). \tag{11}$$

A prior or reference distribution $p_{\psi_i}(z_i \mid c_i)$ constrains the local state. It may be standard normal, contextual, hierarchical, or temporal [13,36,45]. The concrete implementation name used in the controlled experiments is EVE (*Elemental Variational Expanse*). The mechanistic tests use its nonlinear $k = 2$ configuration; the fresh exact-capacity follow-up uses a separately declared Gaussian-affine $k = 1$ configuration that preserves the local latent object while matching the comparator's Gaussian predictive family.

For a set $\mathcal{I}$ of declared units, a common end-to-end objective is

$$\mathcal{J}_{\text{local}} = \mathbb{E}_\epsilon[\ell_{\text{task}}(f(x; z), y)] + \sum_{i \in \mathcal{I}} \beta_i \,\mathrm{KL}(q_{\phi_i}(z_i \mid h_i, c_i) \| p_{\psi_i}(z_i \mid c_i)) + \Omega, \tag{12}$$

where $\beta_i \geq 0$ weights the local KL contribution of unit i, and Ω may contain local capacity, homeostatic, temporal, or anti-collapse constraints. In the principal nonlinear $k = 2$ benchmark configuration, Ω is instantiated explicitly as a soft activity-band penalty; the Gaussian-affine exact-capacity refinement fixes the local latent mean to zero and does not use this band. For a minibatch B, define the mean latent-energy statistic

$$\widehat{m}_i(B) = \frac{1}{|B|} \sum_{h \in B} \|\mu_{\phi_i}(h, c_i)\|_2^2, \tag{13}$$

and

$$\Omega_{\text{band}} = \lambda_{\text{band}} \sum_{i \in \mathcal{I}} \left[(\ell_i - \widehat{m}_i)_+^2 + (\widehat{m}_i - u_i)_+^2 \right], \qquad (r)_+ = \max(0, r). \tag{14}$$

The mechanistic experiments use $\lambda_{\text{band}} = 0.05$ and $[\ell, u] = [0.01, 2.0]$ for the nonlinear $k = 2$ EVE variational neuron. The Gaussian-affine exact-capacity refinement is an explicitly stated exception with fixed zero latent mean and no activity-band term. For a conditionally factorized layer,

$$\begin{aligned} q_\Phi(\mathbf{z} \mid \mathbf{h}, \mathbf{c}) &= \prod_{i \in \mathcal{I}} q_{\phi_i}(z_i \mid h_i, c_i), \\ \mathcal{K}_{\text{layer}} &= \sum_{i \in \mathcal{I}} \mathrm{KL}(q_{\phi_i}(z_i \mid h_i, c_i) \,\|\, p_{\psi_i}(z_i \mid c_i)) \,. \end{aligned} \tag{15}$$

This factorization represents multiple local variational processes. Units usually output a point activation from a sample or declared statistic; propagation of distribution parameters must be declared explicitly.

4.3 Operational computational granularity

Neuron-level granularity is anchored to a declared partition $\mathcal{P} = \{I_1, \ldots, I_J\}$ of local latent coordinates. The boundary below is an operational reporting convention that provides an explicit architectural criterion for the declared computational granularity. Clause (iv) makes unit-level attribution falsifiable by requiring a local computation before downstream mixing, while alternative granularity conventions remain compatible when stated explicitly.

Definition 1 (Unit-level variational specification criterion). *A latent component indexed by i is specified at unit level only when all of the following are part of the architecture specification rather than introduced post hoc for analysis: (i) the model contains an explicitly indexed local variable z_i and conditional family $q_{\phi_i}(z_i \mid h_i, c_i)$; (ii) a prior or reference distribution is declared for that variable; (iii) the unit partition is declared independently of whether the variational family happens to factorize coordinate-wise; (iv) z_i is consumed by a locally scoped map $a_i = F_{\eta_i}(h_i, z_i)$ before downstream mixing, so the random state has a declared local computational role rather than merely being one*

coordinate of a jointly consumed representation-level latent; (v) the inferential or regularization contribution attached to that declared unit is separately computable from the stated model factorization, even when parameters are shared; (vi) diagnostics such as local scale, divergence, mean energy, or utilization are reported at the same predeclared unit index; and (vii) latent dimension and parameter-sharing conventions are part of the model specification.

Worked near-miss. Consider an architecture that predeclares indexed Gaussian blocks z_i, assigns each a reference distribution, exposes a decomposable local KL and indexed diagnostics, and fixes latent dimensions and parameter sharing before analysis, but concatenates all z_i and sends them directly to one shared decoder $G(h, \mathbf{z})$ with no locally scoped $F_{\eta_i}(h_i, z_i)$ before mixing. It satisfies clauses (i)–(iii) and (v)–(vii) but fails clause (iv), so it is *not* a collection of variational neurons under Definition 1. Thus indexing, coordinate-wise factorization, and reportability are insufficient under this convention. A contrasting partial-coupling case is given in Appendix A: parameter sharing and downstream mixing are allowed once each z_i has first entered its declared local map.

A k-dimensional latent vector is one unit when its distribution, reference, cost, local forward role, and diagnostics are jointly defined. It represents k scalar units only when the architectural partition and coordinate-wise or block-wise local maps are explicitly declared before analysis and the corresponding quantities are separately reportable. An inseparable joint latent consumed only by a shared decoder is therefore an example-, representation-, block-, or layer-level latent rather than a collection of variational neurons. Under the terminology of this article, a declared hidden unit satisfying Definition 1 is a *variational neuron*; EVE is the specific implementation name used here, not the name of the criterion itself.

Definition 2 (Partition-level unit-wise variational criterion). *A construction has unit-level variational organization under a stated partition if and only if every declared unit satisfies Definition 1.*

Here, *neuron* refers to the declared computational granularity of the model, independently of its software implementation.

Architectural consequences of the declared partition. Under Eq. (15), let $\mathcal{K}_i = \mathrm{KL}(q_{\phi_i}(z_i \mid h_i, c_i) \| p_{\psi_i}(z_i \mid c_i))$, with $\mathcal{K}_{\text{layer}} = \sum_{i \in \mathcal{I}} \mathcal{K}_i$. Within an architecture that already satisfies Definition 1, the shared index i aligns latent state, local cost, local forward dependence, and diagnostics, enabling unit-wise decomposition even with parameter sharing. This is an accounting consequence of the declared architecture, not evidence that the local state is functionally important or semantically identified. Equation (11) also defines three local policies: $z_i \sim q_{\phi_i}$, $z_i = \mu_{\phi_i}$, and $z_i \sim p_{\psi_i}$, corresponding to learned-state sampling, mean-state execution, and reference-state intervention.

The corresponding computation is written as explicit pseudocode in Appendix C; Eqs. (9)–(12) contain the full mathematical specification.

4.4 Strict posterior inference and local variational regularization

Two regimes must be distinguished.

Strict probabilistic reading. There is an explicit observed or declared local target a_i^*, with a local likelihood or factor $p(a_i^* \mid z_i, h_i, c_i)$, and a joint model under which $q_{\phi_i}(z_i \mid h_i, c_i, a_i^*)$ approximates a posterior. For unit coefficient one,

$$\mathcal{L}_i = \mathbb{E}_{q_{\phi_i}}[\log p_{\eta_i}(a_i^* \mid z_i, h_i, c_i)] - \mathrm{KL}(q_{\phi_i} \| p_{\psi_i}). \tag{16}$$

This is a genuine local ELBO. A coefficient $\beta_i \neq 1$ gives a weighted variational objective.

End-to-end architectural reading. The unit has no independent target a_i^*. Its local state is composed by later layers and trained through the downstream task likelihood. The data-fit term is global while the KL contribution is local. Equation (12) is then described as *locally factorized variational regularization inside a globally trained stochastic network*. The term *posterior* should be reserved for a declared posterior target.

Accordingly, the phrase *unit-level variational latent inference* is appropriate when an explicit latent-variable model defines the posterior target. When the same local distribution is trained only through a downstream task plus a local reference penalty, the more precise generic description is *unit-level variational regularization*; no posterior semantics are implied.

4.5 Deterministic policy, dimensionality, and observability

At the forward-computation level, $\sigma_{\phi_i}(h_i, c_i) \to 0$ concentrates the local distribution toward a point state. This is not generally a finite-objective limit under a fixed non-degenerate prior because $\mathrm{KL}(\mathcal{N}(\mu, \sigma^2) \| \mathcal{N}(0, I))$ diverges as $\sigma \to 0$. In the principal nonlinear $k = 2$ EVE configuration, $\beta = 10^{-3}$, encoded log-variance clipping $[-7, 3]$, and the mean-energy band $\lambda_{\mathrm{band}} = 0.05$, $[\ell, u] = [0.01, 2.0]$ are fixed protocol safeguards; the band constrains $\|\mu\|^2$, not σ^2. The Gaussian-affine exact-capacity refinement instead fixes $\mu = 0$, parameterizes local scale by softplus, omits the band, and retains $\beta = 10^{-3}$. These choices provide protocol control, while non-degeneracy is assessed empirically through the reported local diagnostics and interventions. Separately, replacing a sampled state by its mean defines a deterministic deployment policy even when the learned distribution remains non-degenerate.

The latent dimension k_i is not network width. Width counts units; k_i counts local latent degrees of freedom. The atomic case $k_i = 1$ defines one stochastic coordinate, whereas $k_i > 1$ defines a local latent subspace.

Suppressing contextual arguments for readability, let $\Sigma_i = \mathrm{Diag}(\sigma_i^2)$ denote the local covariance. Useful diagnostics include

$$\mathcal{K}_i, \qquad \|\mu_i\|^2, \qquad \mathrm{tr}\, \Sigma_i, \qquad I(z_i; y \mid h_i), \tag{17}$$

where $I(z_i; y \mid h_i)$ denotes conditional mutual information when such a quantity is estimated. These quantities can characterize local activity or utilization; they do not by themselves prove a particular uncertainty semantics or general predictive advantage.

4.6 When is a variational neuron Bayesian?

The experiments in Section 6 instantiate a variationally regularized stochastic architecture with point-estimated weights: each declared unit uses $q_{\phi_i}(z_i \mid h_i)$ and a KL penalty to a reference distribution, and prediction may use samples or means. This defines Level 1 in the taxonomy. Bayesian inference over local latent variables and a hierarchical construction that also places priors over weights define Levels 2 and 3, whose joint densities and posterior-predictive equations are formalized in Appendix C.

5 Comparison with Related Probabilistic Architectures

The variational-neuron construction considered here is characterized by the joint specification of a local random object, conditioning, reference distribution, decomposable cost, forward role, and indexed diagnostics. EVE denotes the concrete implementation evaluated in the experiments. Section 1.1 states the delta from the directly overlapping earlier work [41,42]. The present contribution is the unit-level specification and partition rule, the four-axis/six-question reporting framework, the explicit posterior-claim rule, the evidence hierarchy, and the controlled empirical suite.

The nearest neighbors differ primarily in random object and computational location. Representative examples include example-level latent-variable models and variational bottlenecks [27,3], weight-space VI with local reparameterization [26], output and final-layer probabilistic methods [25,63], and probabilistic Transformers [60]. Shared stochastic or variational ingredients do not by themselves satisfy the predeclared unit-local routing criterion of Definition 1. Appendix A retains the detailed crosswalk, including VRNNs, probabilistic embeddings, distribution-valued state propagation, and layer-wise Transformer latents.

6 Controlled Empirical Tests

The empirical suite provides a controlled worked application of the reporting framework on one synthetic regression family and one EVE architecture family. Four mechanistic experiments use ten paired data splits; the exact-capacity output comparison is a locked fresh-seed follow-up using 50 paired splits. The experiments instantiate structural specification, fixed-checkpoint functional use, latent dimensionality, controlled noise response, and predictive comparison after parameter-count matching.

6.1 Common protocol and compared architectures

The data-generating process is the one-dimensional heteroscedastic regression problem in Appendix B. Each seed uses independently generated training, validation, and test sets of 2,500, 900, and 1,500 observations. Models are trained with Adam, learning rate 2×10^{-3}, gradient clipping at norm 5, validation every 25 epochs, and early stopping after a minimum of 500 epochs. The nonlinear $k = 2$ experiments use Monte Carlo sampling as specified in Appendix B; the Gaussian-affine exact-capacity refinement has an analytic Gaussian marginal and is scored without Monte Carlo approximation.

The principal mechanistic model is the nonlinear $k = 2$ EVE variational neuron: it has a diagonal-Gaussian latent inside a hidden encoder–decoder cell, standard-normal reference, point-estimated weights, $\beta = 10^{-3}$, and the latent-energy range penalty in Eqs. (13)–(14); it contains 2,472 trainable parameters. The fresh exact-capacity follow-up uses a separately declared Gaussian-affine $k = 1$ EVE configuration with $q(z \mid h) = \mathcal{N}(0, s_\phi(h)^2)$ and local map $a = m_\eta(h) + gz$, also with 2,472 parameters. Its $\beta = 10^{-3}$ setting was selected on validation seeds 40–59 and locked before evaluation on fresh seeds 400–449. Comparator details are in Appendix B.

6.2 Results summary

All detailed paired tables, confidence intervals, intervention definitions, figures, per-seed values, and architecture-specific results are retained in Appendix B; the main text reports the inferentially decisive quantities. Here, *resolved*/*unresolved* means that the paired 95% CI excludes/includes zero; this is descriptive, not a decision rule. Table 2 summarizes one representative quantity from each evidence level while preserving the interpretation boundaries.

The exact-capacity comparison isolates predictive behavior under matched parameter count and predictive family. The final Gaussian-affine EVE and heteroscedastic output model each contain 2,472 trainable parameters and are evaluated on 50 fresh paired splits after validation-only locking. Paired MSE, NLL, CRPS, and MACE intervals all include zero: -0.0000046 $[-0.0000331, 0.0000240]$, -0.000098 $[-0.000494, 0.000298]$, -0.0000095 $[-0.0000394, 0.0000204]$, and -0.000031 $[-0.000443, 0.000381]$, respectively. The corresponding Wilcoxon p-values are 0.848, 0.383, 0.466, and 0.947. Collectively, the paired estimates show closely matched predictive performance on this controlled family at the attained sensitivity.

At $n = 50$, the attained 80%-power sensitivity, computed from the observed paired standard deviations under a two-sided $\alpha = 0.05$ normal approximation, is 4.0×10^{-5} for MSE, 5.5×10^{-4} for NLL, 4.2×10^{-5} for CRPS, and 5.7×10^{-4} for MACE. The observed paired effects are smaller than these thresholds, providing a quantitative resolution context for the closely matched predictive results.

Across the six nominal calibration levels in the fresh exact-capacity comparison, the paired MACE difference is -0.000031 (95% CI $[-0.000443, 0.000381]$), with its interval including zero; the 90% coverage/width values in Appendix B

Table 2. Selected descriptive results from the illustrative EVE worked example. Complete numerical tables and figures are in Appendix B. Differences are EVE minus comparator or intervention minus ordinary EVE sampling as stated.

Test	Key numerical result	Supported interpretation
Deterministic hidden cell	ΔMSE $= -0.000196$ $[-0.000534, 0.000142]$; ΔNLL $= -0.451$ $[-0.459, -0.443]$	Probabilistic behavior differs while point accuracy remains closely matched.
Fresh exact 2,472-parameter output match ($n = 50$)	all four primary paired CIs include zero; representative NLL $\Delta = -0.000098$ $[-0.000494, 0.000298]$	Predictive performance is closely matched at the attained sensitivity, with all four primary paired intervals including zero.
Fixed-checkpoint latent intervention	full-state shuffle: ΔMSE $= 0.0994$ $[0.0631, 0.136]$; variance-only shuffle: ΔMSE $= -0.000147$ $[-0.000340, 0.0000460]$, ΔNLL $= 0.0842$ $[0.0297, 0.139]$	Functional use and variance-specific probabilistic sensitivity at fixed weights.
Latent dimension $k \in \{1, 2, 4, 8\}$	predictive intervals versus $k = 2$ include zero; at $k = 8$, Δ KL participation ratio $= +3.25$ $[2.57, 3.93]$	The participation ratio quantifies how local KL allocation is distributed across latent coordinates, complementing predictive metrics.
Controlled noise relocation	center-tracking slope $= 1.13$ $[0.975, 1.29]$; mean center correlation $= 0.992$ (approximately $[0.985, 0.998]$)	Learned covariance tracks the controlled relocation of observation variability.

provide representative interval diagnostics. Fixed-checkpoint interventions answer a complementary question. Shuffling the complete local latent state changes point prediction, whereas shuffling only the input-conditioned variance leaves MSE near zero while changing NLL, CRPS, coverage, and interval width. The k sweep quantifies descriptive latent allocation across dimensions. Relocating the conditional-noise peak produces corresponding covariance movement; for slope 1.13, unit slope lies inside the 95% CI, consistent with unit tracking. Together, these tests instantiate the evidence hierarchy across structural, functional, causal, semantic, and predictive levels within the controlled EVE setting.

6.3 Empirical interpretation

The five experiments address complementary questions: structural specification, functional utilization, same-checkpoint causal relevance, descriptive latent allocation, controlled sensitivity to known observation variability, and capacity-controlled predictive utility. Their combined role is methodological: they instantiate how progressively stronger claims can be evaluated separately and reported at their corresponding evidence level for one EVE configuration.

7 Discussion and Generic Reporting Recommendations

7.1 What does a variational neuron represent?

Before explicit identification, $\sigma_i^2(h_i)$ and $\mathrm{KL}_i(h_i)$ should be described as *local latent dispersion*, *local variational activity*, or *latent representation variability*. A

local state may react to observation noise, ambiguity, distribution shift, model conflict, capacity limits, or optimization pressure. The words *epistemic* and *aleatoric* require additional modelling assumptions or empirical identification.

A hybrid model can separate variation over weights from variation over local states:

$$\widetilde{p}_q(y \mid x, \mathcal{D}) := \iint p(y \mid x, w, z) q_\lambda(w \mid \mathcal{D}) q_\phi(z \mid x, w)\, dz\, dw. \tag{18}$$

Variation across w samples concerns the approximate posterior over models; variation across z samples concerns the local state conditional on the input and model.

An aleatoric interpretation is available when the latent participates in an observation model such as

$$z \sim p_\psi(z \mid x), \qquad y \sim p_\theta(y \mid x, z), \tag{19}$$

and the resulting conditional variability is identified relative to the likelihood. Calibration remains separate from semantics and should be evaluated through proper scoring rules, coverage, risk–coverage analysis, and robustness under shift [17,35].

7.2 Structural specification and evidence levels

Definition 1 answers the structural question: *is there a separately specified local variational object at the declared computational index?* Functional utilization, causal effect, uncertainty semantics, and predictive benefit are then addressed through the corresponding evidence levels.

The reporting hierarchy has five levels: (A) structural specification, (B) functional utilization, (C) same-checkpoint causal relevance, (D) semantic identification, and (E) predictive utility. These are non-equivalent claims. Appendix A retains the complete evidence table mapping each level to the evidence available in this study.

The hierarchy aligns each empirical result with its evidential role. Same-checkpoint interventions directly support Levels B and C on the synthetic benchmark. Controlled noise relocation provides Level D evidence that local covariance is responsive to the manipulated source of observation variability within this controlled setting. The exact-capacity comparison provides a Level E statement calibrated to the stated data-generating process.

7.3 Terminological rule

The taxonomy uses established field terminology. The generic term *variational neuron* denotes a declared hidden unit satisfying Definition 1, while EVE is retained as the implementation name used in the worked case study. Standard field-wide acronyms retain their usual meanings. Appendix A provides the complete vocabulary and worked mappings.

7.4 Scope and evidential context

The empirical scope comprises one synthetic 1-D regression family, one EVE architecture family, ten-seed mechanistic analyses, and a 50-seed exact-capacity follow-up. A complementary structural application records Definition 1 clause by clause for four published third-party architectures, spanning example-, representation-, weight-, and output-level variational loci. The nonlinear $k = 2$ EVE configuration fixes $\beta = 10^{-3}$, band coefficient 0.05, bounds $[0.01, 2.0]$, and log-variance clipping $[-7, 3]$; the Gaussian-affine $k = 1$ refinement fixes latent mean to zero, omits the band, and retains $\beta = 10^{-3}$. The Gaussian-affine setting was selected on validation seeds 40–59 before evaluation on fresh seeds 400–449, with earlier exploratory runs informing the architectural refinement. Together, these design choices define a transparent evidential context for the structural and empirical claims reported here.

8 Conclusion

Bayesian neural networks and variational neurons can reuse probabilistic devices while differing in random object and computational location. This paper contributes the four-axis/six-question framework, Definition 1, the posterior-claim rule, and the evidence hierarchy; EVE provides the worked implementation. Clause-level applications to published VAE, VIB, locally reparameterized BNN, and VIFO architectures show how the criterion distinguishes example-, representation-, weight-, output-, and unit-level probabilistic organization from architectural descriptions. In the EVE case study, fixed-checkpoint interventions provide benchmark-specific evidence of functional sensitivity; variance-only shuffling changes probabilistic scores while MSE remains near zero; latent-dimensional analyses characterize internal allocation; relocated noise is tracked by learned covariance; and fresh exact-capacity differences in MSE, NLL, CRPS, and MACE show closely matched predictive performance at the attained sensitivity. Together, the structural applications and controlled experiments demonstrate the framework's use for separating structural, functional, causal, semantic, and predictive claims, while Bayesian Levels 2 and 3 provide formal extensions of the taxonomy.

Data and Code Availability Protocol and numerical results are in Section 6 and Appendix B; code and generated outputs are in the project repository, and the archival article is at Zenodo DOI 10.5281/zenodo.21870089.

Disclosure of Interests. The author declares no competing interests relevant to this article.

A Extended Positioning of Related Architectures

A.1 Contribution map

Table 3. Established foundations and the contribution formalized in this article.

Element	Established foundation	Contribution formalized here
Probabilistic taxonomy	Bayesian deep-learning and uncertainty-quantification surveys	Four non-equivalent axes combined with six operational questions that identify the random object and its computational path
Local latent computation	variational autoencoders (VAEs), deep latent-variable models, stochastic hidden units, and information bottlenecks	A unit-level specification criterion with declared conditioning, local reference distribution, decomposable variational cost, explicit forward role, and unit-indexed diagnostics
Controlled case study	Deterministic, heteroscedastic, and stochastic regression benchmarks	Ten-seed mechanistic tests plus a fresh 50-seed exact-capacity output comparison on one controlled regression family

A.2 Relation to broad UQ taxonomies

Table 4 makes the novelty boundary explicit. The comparison emphasizes complementary organizing dimensions across established UQ syntheses and the architecture-reporting framework used here.

Table 4. Organizing emphasis of representative UQ syntheses and the complementary architecture-reporting protocol used here.

Source	Primary organizing emphasis	Relation to this article
Gawlikowski et al. [17]	Sources/types of uncertainty, estimation-method families, uncertainty measures, calibration, benchmarks, and applications	Broad UQ landscape; this article instead requires one architecture instance to report random object, inference/regularization location, conditioning, sampling site, propagated representation, and prediction rule jointly
Hüllermeier and Waegeman [20]	Conceptual and methodological separation of aleatoric and epistemic uncertainty and ways to represent/quantify them	Semantic uncertainty distinction; this article does not replace it, but separates semantic identification from structural placement and functional/causal evidence
This article	Computational object/location plus staged claim–evidence reporting	Adds the six-field architecture record, the predeclared unit-local routing convention, and explicit separation of structure, use, fixed-checkpoint causal relevance, semantics, and predictive utility

A.3 Evidence hierarchy

Table 5. Evidence levels that should be reported separately.

Level	Question	Evidence in this article
A. Structural specification	Is a local variational object architecturally defined at unit index i before analysis?	Yes: Definition 1, predeclared partition, locally scoped map, local family/reference, cost, and indexed diagnostics
B. Functional utilization	Is the local state actually used by the trained computation?	Yes on the controlled benchmark: reference, zero, mean-state, and shuffled-state interventions change downstream behavior
C. Same-checkpoint causal relevance	Does changing the local state while holding all learned weights fixed change predictions or predictive uncertainty?	Yes on the controlled benchmark: full-state shuffling changes MSE and probabilistic scores; variance-only shuffling changes probabilistic scores without a resolved MSE effect
D. Semantic identification	Does the local distribution represent a named source such as aleatoric or epistemic uncertainty?	Controlled noise relocation provides benchmark-specific semantic sensitivity under the known data-generating process
E. Predictive utility	Does the mechanism improve task performance, calibration, or proper scores?	The fresh 50-seed exact-capacity comparison shows closely matched MSE, NLL, CRPS, and MACE at the attained sensitivity

A.4 Clause-level application to published architectures

To exercise Definition 1 outside the EVE design, Table 6 applies the seven clauses directly to four published architectures selected for distinct probabilistic loci: a standard VAE [27], Deep VIB [3], a variational BNN using local reparameterization [26], and VIFO [63]. The record follows each architecture's published random object, reference or prior, computational partition, routing, variational cost, diagnostics, and dimensional specification. The resulting labels are determined by architectural location rather than by the shared presence of KL terms or reparameterized samples.

Table 6. Clause-level application of Definition 1 to published third-party architectures. The middle column records clauses (i)–(vii) in order using the granularity stated by the source architecture.

Architecture	Clause-wise architectural record (i)–(vii)	Resulting generic description
VAE [27]	(i) example-level $q_\phi(z \mid x)$; (ii) declared $p(z)$; (iii) example-level latent vector; (iv) sampled z enters a shared decoder; (v) example-level KL; (vi) latent-level diagnostics; (vii) latent dimension specified	Variational latent-variable model at example granularity
Deep VIB [3]	(i) input-conditioned bottleneck distribution; (ii) declared reference $r(z)$; (iii) representation-level bottleneck; (iv) sampled bottleneck enters the predictor; (v) bottleneck KL; (vi) representation-level diagnostics; (vii) bottleneck dimension specified	Variational bottleneck at representation granularity
Local-reparameterization BNN [26]	(i) inferential object is the weight posterior $q(w)$; (ii) weight prior; (iii) weight-space factorization; (iv) sampled pre-activations provide a local estimator; (v) weight-space KL; (vi) weight/posterior diagnostics; (vii) weight shapes and sharing specified	Variational Bayesian neural network with local sampling and weight-space inference
VIFO [63]	(i) final-layer/output-space variational variable; (ii) Bayesian variational formulation at the predictive interface; (iii) output-space organization; (iv) probabilistic computation occurs at the final-layer interface; (v) final-layer variational objective; (vi) output-level diagnostics; (vii) output dimension specified	Final-layer variational method at predictive-interface granularity

These four applications span distinct loci while using the same clause order and reporting fields. The format therefore provides an auditable architecture record that can be reproduced from published descriptions and compared directly across independent annotations.

A.5 Boundary cases for Definition 1

The local-routing clause is intentionally a convention about computational order, not parameter ownership. For a partial-coupling architecture in which each z_i first enters $a_i = F_{\eta_i}(h_i, z_i)$ and only the resulting a_i are subsequently mixed, clause (iv) is satisfied even if all F_{η_i} share parameters. By contrast, if the latent blocks are mixed into a joint representation before any unit-indexed map, the construction falls on the representation/block side of the declared boundary. These two cases make the operational boundary explicit while remaining compatible with alternative terminology when its granularity convention is declared.

A.6 Detailed comparison with neighboring architectures

The following subsections apply the same random-object and computational-location distinctions to representative neighboring families; the goal is comparative positioning rather than reclassification by acronym.

Latent-variable models, stochastic units, and bottlenecks VAEs and stochastic backpropagation established amortized inference for input-conditioned latent variables [27,39]. Deep exponential families, variational recurrent neural networks, and ladder VAEs distribute latent variables across depth or time [13,36,45]. Their latents are usually organized by example, layer, hierarchy, or sequence. A variational neuron uses analogous machinery at a declared unit or small block.

Stochastic hidden units predate modern VI and include randomized activations, gates, and conditional-computation variables [7,23,55,30,46,48]. Sampling a hidden variable is not sufficient for the variational-neuron label: the local distribution, reference, cost, conditioning, and forward role must also be identified.

The deep variational information bottleneck, Information Dropout, and the Markov Information Bottleneck provide especially close antecedents through input-conditioned stochastic representations and decomposable KL-like constraints [2,3,59]. Probabilistic embeddings similarly represent inputs by distributions [34,44]. Coordinate-wise KL factorization alone does not move these methods to neuron level. They qualify as unit-level variational-neuron equivalents only when a predeclared architectural partition routes the corresponding local latent states through locally scoped computations satisfying Definition 1; otherwise they remain representation-, layer-, or embedding-level constructions.

Weight uncertainty, dropout, and local reparameterization Variational BNNs place approximate posteriors over weights [11,18]. The local reparameterization trick may sample pre-activations, but the inferential random object is still the weight posterior [26]. Dropout can be interpreted as regularization, model averaging, or, under particular assumptions, approximate variational Bayesian inference [16]. In all cases, the object specified by the joint model takes precedence over the location at which equivalent noise is sampled.

Output distributions and final-layer inference Heteroscedastic regression predicts parameters of an observation distribution and is distributional at the predictive interface rather than inside the hidden unit [25]. Variational Inference on the Final-Layer Output (VIFO) and Variational Classification place variational inference at the final-layer or pre-softmax interface [67,63]. These constructions are close comparators because they introduce probabilistic structure near prediction while leaving the hidden-unit location distinct.

Distribution-valued hidden representations and Transformers Assumed-density, expectation-propagation, moment-propagation, and probabilistic backpropagation methods transform approximate beliefs through neural computations [50,53,19,40,61,62]. Analytic covariance propagation, Gaussian-mixture propagation, output-distribution bounds, and probabilistic state-space layers extend this family [52,56,57,58,64]. Their defining question

is what distributional representation crosses layers. By contrast, a variational neuron may produce an ordinary point-valued activation after using its latent state; the propagated object must be reported rather than inferred from the presence of a distribution inside the unit.

Probabilistic Transformers also occupy several locations. Bayesian Transformer language models and BayesFormer place uncertainty over parameters or use dropout-related approximations [60,65]; Della introduces layer-wise latent variables [54]; variational attention randomizes an attention representation [66]; and Gaussian-process Transformer approaches attach probabilistic semantics to attention kernels or units [51]. None of these labels alone identifies a variational neuron; the six reporting questions remain necessary.

Additional neighboring architectures are summarized in Appendix A (Table 8).

A.7 Recommended generic descriptions

The taxonomy is expressed through established terminology centered on the primary random object, its conditioning, and its computational location. This article retains one model name, EVE, for the specific variational-neuron implementation studied experimentally, while generic architectural descriptions are used for cross-model comparison. Standard field-wide acronyms such as BNN, VI, VAE, KL, ELBO, MSE, NLL, CRPS, and Monte Carlo sampling retain their usual meanings.

A compact generic vocabulary is sufficient; EVE remains the proper name of the implementation studied here:

1. *Bayesian neural network* or *variational Bayesian neural network*: posterior uncertainty over weights or functions, with the inference method named explicitly.
2. *Variational latent-variable model*: an amortized latent distribution approximating a declared posterior under an explicit latent-variable model.
3. *Variational neuron*: a declared hidden computational unit containing an input-conditioned local variational latent variable whose reference distribution, local cost, forward role, and unit-indexed diagnostics are explicitly specified.
4. *Unit-level variational regularization*: the training regime in which such a variational neuron is optimized through a downstream task plus a local reference penalty, without an automatic posterior claim.
5. *Stochastic hidden unit*: randomized hidden computation when no stronger variational or Bayesian semantics are established.
6. *Heteroscedastic output model*: an architecture that predicts input-dependent observation-distribution parameters at the output interface.

Table 7 applies these descriptions to representative architectures.

Table 7. Worked application of the taxonomy using generic, established descriptions.

Example	Primary random object / location	Inference or regularization	Propagated representation / sampling	Generic description
VAE [27]	Example-level latent z	Amortized posterior under a latent-variable model	Sampled z enters a decoder	Variational latent-variable model
VRNN [13]	Time-indexed sequence latent z_t	Sequential amortized VI	Sampled temporal latent conditions recurrent computation	Sequential latent-variable model
VIB [3]	Input-conditioned bottleneck representation	Variational information-bottleneck objective	Sampled representation enters predictor	Variational bottleneck / stochastic representation
MC dropout [16]	Dropout masks; under specific assumptions, approximate model uncertainty	Regularization or approximate VI depending on assumptions	Masked operators produce point activations per realization	Stochastic computation; approximate Bayesian interpretation only under stated assumptions
Deep ensembles [28]	Whole trained model member	No VI required	Predictions from separate deterministic models are aggregated	Ensemble
BayesFormer [60]	Dropout-induced Transformer parameter/operator uncertainty	Approximate variational-dropout construction	Sampled dropout operators	Approximate Bayesian Transformer
Della [54]	Layer-wise latent variables	Variational latent inference across layers	Layer latents fused with hidden states	Layer-wise variational latent Transformer
VIFO [63]	Final-layer/output-space variable	VI at predictive interface	Probabilistic final representation	Final-layer variational method
Heteroscedastic regressor [25]	Observation-distribution parameters at output	Output likelihood	Deterministic hidden path emits mean and variance	Heteroscedastic output model
EVE (this study)	Input-conditioned latent z_i inside a declared hidden unit	Local reference KL plus downstream task loss	z_i is sampled or replaced by a declared statistic before downstream computation	Variational neuron with unit-level variational regularization

Table 8. Complete positioning relative to neighboring probabilistic architectures. The decisive comparison is the modeled random object, its conditioning, and its location.

Family	Primary object and conditioning	Principal role	Distinction from a variational neuron
Standard BNN	Weights or functions conditioned on dataset $\mathcal{D}$	Posterior model averaging	Global model-level inferential object rather than a per-input local hidden state
Deep or sequential latent model	Global, layered, or temporal latents	Generative modelling or stochastic dynamics	Usually organized at example, layer, hierarchy, or sequence granularity
Stochastic hidden-unit network	Hidden variable or gate sampled during computation	Multimodal prediction or conditional computation	Sampling alone does not establish local variational semantics or decomposable reference cost
Variational bottleneck or probabilistic embedding	Input-conditioned representation distribution	Compression, invariance, or metric learning	Typically representation- or embedding-level unless unit granularity is explicitly declared
Variational BNN with local reparameterization	Weight posterior represented through sampled pre-activations	Efficient gradient estimation	Local sample, but global weight-space random object
Probabilistic state-propagation network	Distributional input, activation, or approximate belief	Layerwise transformation of uncertainty	Focuses on propagated distributional state; local latent identity and cost may differ
Final-layer variational method	Output or pre-softmax latent	Output-space inference and calibration	Predictive-interface location rather than hidden-unit location
Heteroscedastic, mixture, prior-network, or evidential output [8,32,4,43]	Observation distribution or output-parameter distribution	Predictive density or output uncertainty	Distribution appears at the output interface
Ensemble or mixture of experts [28,22]	Whole-model member or routing variable	Model diversity or conditional routing	Whole-model or routing location; compatible but non-equivalent
Distributional reinforcement learning [6]	Return distribution	Distributional value or return prediction	Task-output location rather than a local hidden-state inferential object
Variational neuron (EVE instantiation in this study)	Local latent z_i conditioned on h_i, c_i	Local distributional hidden computation	Declared unit, local reference/cost, forward dependence, and indexed diagnostics

B Controlled Experiment Specification and Evaluation Protocol

B.1 Data-generating process

For each split,

$$x \sim \mathcal{U}[-4, 4], \tag{20}$$

$$y \mid x \sim \mathcal{N}\big(m(x), \sigma^2(x)\big)\,, \tag{21}$$

$$m(x) = 0.85 \tanh(1.35x) + 0.10x, \tag{22}$$

$$\sigma(x) = 0.07 + 0.52 \exp\left[-\frac{1}{2}\left(\frac{x}{0.85}\right)^2\right] + 0.05\left(1 + \exp[-3(|x| - 2.7)]\right)^{-1}. \tag{23}$$

The conditional mean is smooth, with a central high-variance region and mild edge variance. The controlled noise-relocation experiment replaces the center of the main Gaussian variance component by -1.75, 0, or $+1.75$ while leaving the conditional mean unchanged.

B.2 Seeds, optimization, and evaluation

The mechanistic experiments use seeds $s \in \{0, \ldots, 9\}$. The final exact-capacity Gaussian-affine configuration is selected using validation performance on development seeds $s \in \{40, \ldots, 59\}$ and then evaluated on fresh paired seeds $s \in \{400, \ldots, 449\}$ with no test-driven configuration changes. For every seed, independent training, validation, and test sets contain 2,500, 900, and 1,500 observations. Generator seeds are $10000{+}10s{+}1$, $10000{+}10s{+}2$, and $10000{+}10s{+}3$. Models are trained on CUDA-enabled GPUs using Adam with learning rate 2×10^{-3} and gradient clipping at norm 5. Training is capped at 1,200 full-batch epochs; validation is checked every 25 epochs, with early stopping after a minimum of 500 epochs and 12 checks without improvement.

The nonlinear $k = 2$ stochastic experiments use 32 Monte Carlo samples during training, 64 during validation, and 256 during predictive evaluation; their NLL uses a moment-matched Gaussian and CRPS is computed from predictive samples. The Gaussian-affine exact-capacity refinement has an analytic Gaussian marginal, so MSE, Gaussian NLL, Gaussian CRPS, and calibration are evaluated exactly without Monte Carlo scoring noise. Predictive calibration is evaluated at nominal coverages 50%, 60%, 70%, 80%, 90%, and 95%, summarized by mean absolute coverage error (MACE); 90% coverage and width are additionally reported as representative interval diagnostics.

B.3 Architectures

Principal nonlinear EVE variational neuron. The mechanistic EVE model has one scalar deterministic input state feeding a 1–32–32–4 encoder that parameterizes $(\mu, \log \sigma^2)$ for a diagonal-Gaussian latent with $k = 2$. A 3–32–32–1 decoder maps $[z, h]$ to the hidden activation. The reference distribution is $\mathcal{N}(0, I)$,

$\beta = 10^{-3}$, and the latent-energy range penalty has coefficient 0.05 with target range $[0.01, 2.0]$. Encoder log-variance is clipped to $[-7, 3]$. The model has 2,472 trainable parameters.

Gaussian-affine EVE exact-capacity refinement. The fresh output-level control uses a separately declared $k = 1$ EVE configuration with deterministic trunk 1–42–53, a $53 \to 1$ deterministic mean head, and a $53 \to 1$ local scale head. It defines

$$q_\phi(z \mid h) = \mathcal{N}\big(0, s_\phi(h)^2\big), \qquad a = m_\eta(h) + gz, \tag{24}$$

where $s_\phi(h) = \mathrm{softplus}(r_\phi(h)) + 10^{-4}$, $g > 0$ is a learned scalar gain, the reference is $\mathcal{N}(0, 1)$, and $\beta = 10^{-3}$. Hence the declared stochastic unit activation has the analytic marginal $a \mid h \sim \mathcal{N}(m_\eta(h), g^2 s_\phi(h)^2)$. In this one-unit regression head, the scalar prediction is the identity readout of a, so the terminal unit is the declared locus for the exact-capacity control. Predictive scale is carried by the local latent state, and the fixed zero latent mean makes the mean-energy band unnecessary. The trainable count is $(1 \times 42 + 42) + (42 \times 53 + 53) + (53 + 1) + (53 + 1) + 1 = 2{,}472$. The final β setting is locked from validation seeds 40–59 before fresh seeds 400–449.

Auxiliary regularization sensitivity. A separate five-seed, fixed-$k = 2$, hidden-width-32 retraining-time control on the same synthetic family used 500 epochs per configuration and omitted the latent-energy band penalty, contrasting $\beta = 5 \times 10^{-2}$ with $\beta = 10^{-4}$. Mean local KL changes from 0.056 to 4.280 and mean latent energy $\|\mu_i\|^2$ from 0.029 to 7.909, while test MSE is 0.0776 versus 0.0779, CRPS 0.1275 versus 0.1278, and mean predictive standard deviation 0.2252 versus 0.2275. This retraining-time control characterizes sensitivity to β under the stated auxiliary configuration; same-checkpoint causal sensitivity is addressed separately by the fixed-checkpoint intervention protocol. The result shows that large β-driven changes in the internal variational regime can coexist with similar output metrics.

Capacity-matched deterministic hidden cell. The deterministic hidden-cell comparator uses a 1–32–32–4 encoder and a 5–32–32–1 decoder and has 2,536 trainable parameters. It uses a learned global Gaussian observation variance but has no local latent variable or local KL term.

Standard heteroscedastic output comparator. The standard output comparator has 1,158 parameters. Its hidden computation is deterministic; an input-dependent variance head predicts the observation scale at the output interface. This model serves as the preliminary output-location comparator, while the matched 2,472-parameter model below provides the exact-capacity comparison.

Exact-capacity heteroscedastic output comparator. The output model uses the same deterministic trunk 1–42–53, two output coordinates for predictive mean and raw scale, and one learned global raw-scale offset. Its count is $(1 \times 42 + 42) + (42 \times 53 + 53) + (53 \times 2 + 2) + 1 = 2{,}472$, exactly matching the Gaussian-affine EVE refinement.

B.4 Paired analysis

All headline effects are computed seed-wise before aggregation. For a metric M and paired models A and B,

$$d_s = M_{A,s} - M_{B,s}, \qquad \bar{d} = \frac{1}{n}\sum_{s=1}^{n} d_s. \tag{25}$$

Two-sided 95% Student-t intervals are reported for $\bar{d}$ as uncertainty summaries for the paired effects. The mechanistic experiments use $n = 10$; the fresh exact-capacity comparison uses $n = 50$ and additionally reports two-sided Wilcoxon signed-rank checks and deterministic paired-bootstrap intervals.

*Fresh-*50 *sensitivity.* Using the observed paired standard deviations and a normal approximation for a two-sided $\alpha = 0.05$ test at 80% power, the fresh-50 sensitivity corresponds to mean paired differences of approximately 4.0×10^{-5} for MSE, 5.5×10^{-4} for NLL, 4.2×10^{-5} for CRPS, and 5.7×10^{-4} for MACE. The observed point estimates are smaller than these values. This calculation quantifies the attained resolution and situates the closely matched aggregate predictive results within it.

B.5 Fixed-checkpoint intervention protocol

For each seed, one $k = 2$ EVE variational neuron is trained once. Without modifying any learned weights, the test set is reevaluated under six policies: ordinary local-distribution sampling, local mean-state execution, standard-normal reference sampling, zero latent state, complete latent-state permutation across inputs, and input-conditioned variance permutation across inputs. Common latent and observation random numbers are reused wherever the policy permits. This design isolates downstream sensitivity to the latent mechanism at a fixed checkpoint.

B.6 Latent-dimensionality protocol

Latent dimensions $k = 1, 2, 4, 8$ are evaluated with hidden widths selected to keep capacity close to the $k = 2$ reference. The resulting parameter counts are 2,514, 2,472, 2,523, and 2,456. In addition to predictive metrics, the analysis records total KL, KL per latent dimension, latent mean energy, covariance trace, and a KL participation ratio computed from mean per-dimension KL contributions. The participation ratio serves as a descriptive allocation statistic, while task relevance is assessed separately through predictive metrics and interventions.

B.7 Controlled noise-relocation protocol

For each seed, three datasets reuse the same input samples and standardized Gaussian noise draws while moving only the center of the dominant conditional-noise component. The learned covariance trace is evaluated on a fixed input grid.

Table 9. Ten-seed paired effects of the $k = 2$ EVE variational neuron. Differences are EVE minus comparator; negative values favor EVE for MSE, NLL, and CRPS.

Comparator	Metric	Paired Δ	95% Student-t CI
Capacity-matched deterministic hidden cell (2,536 parameters)	MSE	−0.000196	[−0.000534, 0.000142]
	NLL	−0.451	[−0.459, −0.443]
	CRPS	−0.0152	[−0.0162, −0.0143]
Standard heteroscedastic output model (1,158 parameters)	MSE	−0.00515	[−0.00616, −0.00414]
	NLL	−0.0323	[−0.0367, −0.0278]
	CRPS	−0.00372	[−0.00463, −0.00281]

The analysis reports correlation with the known conditional standard deviation, center of mass of learned versus true variance profiles, and the per-seed slope relating learned noise center to the manipulated true center. Circular shifts of the true variance profile are retained as a secondary descriptive null check alongside the center-tracking analysis.

B.8 Detailed controlled empirical results

This section provides the complete numerical results, confidence intervals, intervention definitions, and supporting figures for the controlled empirical suite.

B.9 Paired comparison with deterministic and output-probabilistic baselines

Table 9 reports seed-paired differences, defined as EVE minus comparator. Negative values favor EVE for MSE, NLL, and CRPS. The deterministic hidden-cell comparison is particularly informative about computational location: the deterministic cell has slightly more parameters (2,536 versus 2,472), yet the paired MSE difference remains unresolved around zero while NLL and CRPS differ strongly.

The first row shows that a change in computational location can alter probabilistic behavior while point-accuracy differences remain near zero. The 1,158-parameter heteroscedastic model provides a preliminary output-level location comparison, and Section B.10 follows with the matched 2,472-versus-2,472 parameter control.

B.10 Exact-capacity output comparison

To control parameter count and predictive family in the output-level comparison, the fresh follow-up compares the standard shared-trunk heteroscedastic output regressor with the Gaussian-affine EVE configuration in Eq. (24). Both have exactly 2,472 trainable parameters and analytic Gaussian predictive marginals, but the random scale is represented at different computational locations: an output scale coordinate in the comparator versus an input-conditioned local

Table 10. Fresh exact 2,472-parameter comparison over 50 paired seeds. Differences are Gaussian-affine EVE minus the heteroscedastic output model; lower is better for MSE, NLL, CRPS, and MACE.

Metric	Paired difference	95% CI	Interpretation
MSE	−0.0000046	[−0.0000331, 0.0000240]	CI includes zero
NLL	−0.000098	[−0.000494, 0.000298]	CI includes zero
CRPS	−0.0000095	[−0.0000394, 0.0000204]	CI includes zero
MACE (50–95%)	−0.000031	[−0.000443, 0.000381]	CI includes zero
Coverage 90%	+0.000480	[−0.000151, 0.001111]	CI includes zero
Width 90%	−0.000854	[−0.001654, −0.000054]	slightly narrower EVE intervals

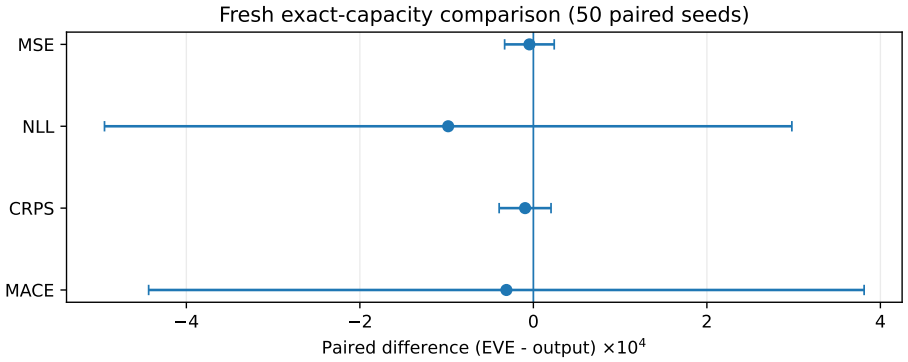

Fig. 2. Paired 95% confidence intervals for MSE, NLL, CRPS, and MACE in the fresh 50-seed exact 2,472-parameter comparison. All four intervals include zero.

latent state in EVE. The configuration is locked on validation seeds 40–59 and evaluated on fresh seeds 400–449.

This fresh control provides a sharper capacity- and predictive-family-matched comparison. With both parameter count and Gaussian predictive family matched, all four primary paired intervals include zero, and the Wilcoxon checks are concordant with those intervals. The result shows that a local variational state and an output-heteroscedastic parameterization can occupy different probabilistic locations while attaining closely matched predictive behavior on this controlled family.

Fresh-seed robustness check. The exact-capacity comparison uses 50 fresh paired seeds. Two-sided Wilcoxon signed-rank p-values are 0.848 for MSE, 0.383 for NLL, 0.466 for CRPS, and 0.947 for MACE, concordant with the paired Student-t intervals. EVE has the lower NLL on 30/50 seeds, while the aggregate paired CI and Wilcoxon result show closely matched aggregate performance at the attained resolution. The secondary predictive-standard-deviation difference is -0.000260 (95% CI $[-0.000503, -0.000017]$), corresponding to slightly sharper

Table 11. Ten-seed paired fixed-checkpoint effects for EVE relative to ordinary sampling from its learned local distribution. Positive ΔMSE, ΔNLL, and ΔCRPS indicate degradation.

Intervention	ΔMSE (95% CI)	ΔNLL (95% CI)	ΔCRPS (95% CI)
Use local mean instead of sampling	0.000114 [−0.000530, 0.000759]	2.82 [2.67, 2.97]	0.0186 [0.0174, 0.0197]
Sample from reference distribution	0.0273 [0.0208, 0.0339]	0.215 [0.158, 0.272]	0.0233 [0.0175, 0.0292]
Shuffle complete latent state across inputs	0.0994 [0.0631, 0.136]	0.713 [0.506, 0.920]	0.0769 [0.0504, 0.103]
Shuffle only input-conditioned latent variance	−0.000147 [−0.000340, 0.000046]	0.0842 [0.0297, 0.139]	0.00166 [0.000267, 0.00306]
Set latent state to zero	0.0196 [0.0151, 0.0241]	3.79 [3.48, 4.10]	0.0415 [0.0353, 0.0478]

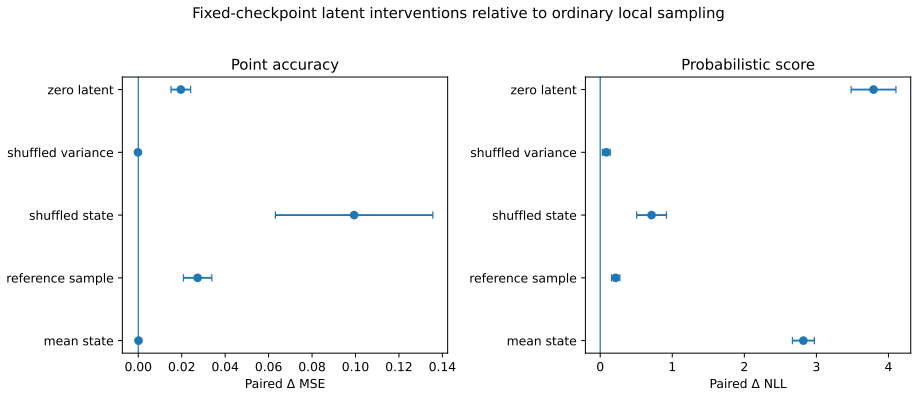

Fig. 3. Selected fixed-checkpoint intervention effects. Shuffling the full latent state changes point accuracy, whereas shuffling only its input-conditioned variance leaves MSE unresolved around zero but degrades NLL. Mean-state execution likewise preserves point accuracy while strongly reducing predictive dispersion.

EVE predictions and reported as a descriptive sharpness diagnostic alongside the primary metrics.

B.11 Fixed-checkpoint interventions

Functional utilization is assessed directly by training each $k = 2$ EVE variational neuron once and evaluating the *same checkpoint* under five interventions while keeping all learned weights fixed. Ordinary sampling from the learned input-conditioned distribution is the reference policy. Common random numbers are used wherever meaningful.

These interventions provide evidence of functional utilization and same-checkpoint causal relevance on this benchmark. They also separate two roles of the local distribution. The latent state as a whole carries information

Table 12. Ten-seed paired differences relative to $k = 2$ under approximately matched capacity. Effective latent dimension is the participation ratio computed from per-dimension KL contributions.

k	Params	ΔMSE	ΔNLL	Δ effective latent dimension
1	2,514	−0.000116 [−0.000625, 0.000394]	0.00193 [−0.000846, 0.00470]	−0.749 [−0.894, −0.605]
2	2,472	reference	reference	reference
4	2,523	−0.000246 [−0.000627, 0.000135]	−0.000500 [−0.00316, 0.00216]	+1.17 [0.669, 1.66]
8	2,456	−0.000379 [−0.000989, 0.000230]	−0.00132 [−0.00417, 0.00153]	+3.25 [2.57, 3.93]

needed for point prediction, as shown by the large MSE increase after shuffling the complete state. The input-conditioned variance, in turn, changes NLL, CRPS, coverage, and interval width while point accuracy remains closely matched. Semantic attribution is examined separately through the controlled noise-relocation experiment.

B.12 Latent-dimensionality analysis

For EVE, the latent dimension k controls the number of local latent degrees of freedom; it is not network width and does not redefine the unit partition. We evaluate $k \in \{1, 2, 4, 8\}$ while adjusting hidden width to keep parameter count within approximately 2.1% of the $k = 2$ model: 2,514, 2,472, 2,523, and 2,456 parameters respectively.

The paired MSE, NLL, and CRPS intervals include zero for $k = 1, 4, 8$ relative to $k = 2$, while total KL and the KL participation ratio change with k. The participation ratio is reported because it quantifies how local KL allocation is distributed across latent coordinates, an internal structural property complementary to predictive metrics. The combined record therefore separates latent-allocation changes from predictive effects. Dimension-wise fixed-checkpoint interventions provide a natural next probe of coordinate-specific functional contributions.

B.13 Controlled relocation of conditional noise

To strengthen identification beyond a single fixed heteroscedastic pattern, we keep the conditional mean fixed while moving the high-variance region to the left, center, or right and test whether EVE's learned local covariance follows that manipulation. For a given seed, the input samples and standardized Gaussian noise draws are held fixed across the three conditions; only the conditional noise scale changes.

Across the three manipulated conditions, the center of mass of the learned local covariance follows the true variance center with a mean per-seed tracking slope of 1.13 (95% CI $[0.975, 1.29]$). The correlation between true and learned centers across the three conditions is also consistently high across seeds (mean 0.992, 95% CI approximately $[0.985, 0.998]$). This manipulation provides benchmark-specific semantic evidence that the learned local covariance is

Table 13. Tracking of the manipulated conditional-noise structure over ten seeds. Correlations compare the learned local covariance trace with the known conditional standard deviation.

Noise location	Post-training correlation	95% CI
Left	0.747	[0.653, 0.842]
Center	0.982	[0.975, 0.989]
Right	0.793	[0.667, 0.918]

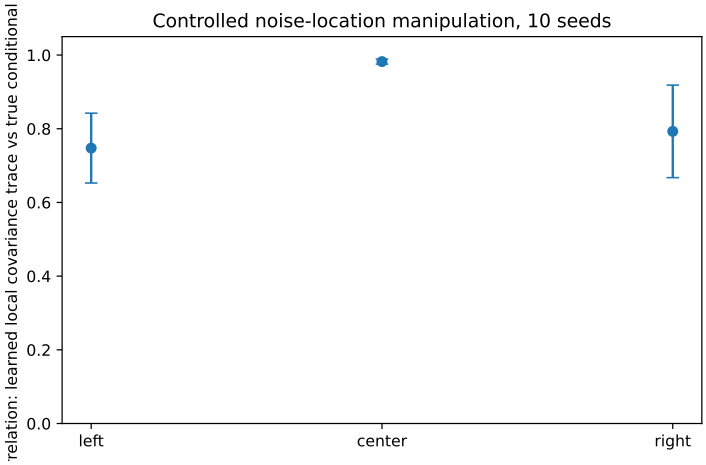

Fig. 4. Post-training correlation between learned local covariance trace and the known conditional standard deviation after moving the high-variance region left, center, and right.

responsive to controlled changes in observation variability under the known synthetic data-generating process.

B.14 Complete headline numerical results

Table 14. Fresh exact-capacity output comparison. Both models contain 2,472 trainable parameters. Values are 50-seed means with 95% confidence intervals for the model-level mean.

Metric	Heteroscedastic output	Gaussian-affine EVE
MSE	0.078282 [0.076700, 0.079864]	0.078277 [0.076697, 0.079858]
NLL	−0.300397 [−0.308316, −0.292478]	−0.300495 [−0.308503, −0.292487]
CRPS	0.127115 [0.125987, 0.128244]	0.127106 [0.125975, 0.128236]
Coverage 90%	0.897867 [0.894959, 0.900774]	0.898347 [0.895200, 0.901493]
Width 90%	0.736140 [0.730387, 0.741893]	0.735286 [0.729632, 0.740940]
MACE (50–95%)	0.011196 [0.009263, 0.013128]	0.011164 [0.009206, 0.013122]

The corresponding paired differences are reported in Table 10. Model-level intervals in Table 14 are descriptive; paired intervals are primary because both models share the same 50 fresh data splits. Calibration uses all six nominal levels (50, 60, 70, 80, 90, and 95%), summarized by MACE. The final $\beta = 10^{-3}$ setting was chosen from a validation-only candidate grid on seeds 40–59 and locked before seeds 400–449.

Table 15: Per-seed paired differences for the fresh exact-capacity comparison. Differences are Gaussian-affine EVE minus the heteroscedastic output model.

Seed	ΔMSE	ΔNLL	ΔCRPS	ΔMACE
400	+0.000193	−0.000366	+0.000039	+0.000222
401	+0.000006	−0.000194	−0.000013	+0.000333
402	−0.000064	+0.000778	+0.000022	−0.000333
403	+0.000027	+0.001074	+0.000082	−0.000444
404	−0.000021	−0.000775	−0.000052	+0.002778
405	+0.000023	+0.001668	+0.000159	+0.000333
406	−0.000110	−0.000592	−0.000105	+0.000778
407	−0.000137	−0.000093	−0.000096	+0.000111
408	+0.000001	+0.000822	+0.000016	−0.000333
409	−0.000000	−0.000214	+0.000046	−0.000556
410	+0.000052	−0.002185	−0.000036	−0.001444
411	−0.000117	−0.003570	−0.000297	+0.001889
412	−0.000055	−0.001276	−0.000054	−0.000111
413	−0.000048	−0.001464	−0.000062	+0.000778
414	+0.000044	+0.000431	+0.000021	−0.001000
415	+0.000009	−0.000247	+0.000002	−0.004333
416	+0.000049	+0.000230	+0.000007	+0.000111
417	+0.000159	+0.003420	+0.000227	+0.002111
418	−0.000170	−0.000870	−0.000125	+0.001111
419	+0.000039	−0.002721	−0.000102	−0.001333
420	+0.000030	+0.000230	+0.000023	+0.000889
421	−0.000021	+0.001490	+0.000081	+0.000556
422	+0.000063	−0.000224	+0.000011	−0.000444
423	−0.000169	+0.000882	−0.000036	−0.001222
424	−0.000046	−0.000392	−0.000030	−0.001444
425	+0.000027	−0.000162	+0.000009	+0.000333
426	−0.000032	−0.000057	−0.000037	−0.002333
427	+0.000020	+0.000417	+0.000049	+0.001444
428	−0.000146	−0.001260	−0.000089	+0.000222
429	+0.000087	−0.001152	−0.000001	−0.000667
430	+0.000124	−0.000506	+0.000001	+0.000556
431	−0.000240	−0.002513	−0.000282	+0.000889
432	+0.000152	+0.000048	+0.000069	−0.000222

Table 15 – continued

Seed	ΔMSE	ΔNLL	ΔCRPS	ΔMACE
433	−0.000032	+0.000052	−0.000059	−0.001667
434	−0.000028	+0.002484	−0.000004	−0.002333
435	+0.000080	−0.000533	+0.000058	+0.001889
436	+0.000129	+0.002152	+0.000241	+0.001222
437	−0.000135	−0.000531	−0.000105	−0.001111
438	+0.000097	+0.001141	+0.000097	+0.004111
439	+0.000004	−0.000256	−0.000004	+0.000000
440	−0.000046	+0.001980	−0.000049	−0.000667
441	−0.000037	−0.000664	+0.000016	+0.000000
442	+0.000023	+0.000114	+0.000017	+0.001222
443	+0.000086	+0.003016	+0.000174	+0.001667
444	−0.000038	−0.000668	−0.000075	−0.000000
445	+0.000223	+0.000655	+0.000133	−0.001333
446	−0.000031	−0.001745	−0.000056	+0.000333
447	−0.000247	−0.002141	−0.000245	−0.000667
448	−0.000010	−0.000324	−0.000024	−0.002667
449	+0.000005	−0.000299	−0.000042	−0.000778

Across the 50 fresh pairs, the two-sided Wilcoxon signed-rank p-values are 0.848 for MSE, 0.383 for NLL, 0.466 for CRPS, and 0.947 for MACE. The numbers of negative/positive paired differences are 24/26, 30/20, 26/24, and 24/26, respectively. The paired-bootstrap 95% intervals likewise include zero for all four primary metrics. These results replace the earlier small-seed exact-capacity estimate for the final reported configuration.

Local-state and oracle diagnostics for the locked refinement. Across the 50 fresh EVE runs, mean local KL is 0.505 ± 0.032, mean q-scale is 1.005 ± 0.026, learned gain is 0.2225 ± 0.0030, and mean effective predictive standard deviation is 0.22351 ± 0.00605. On the ten development seeds 40–49 used for an oracle diagnostic against the known synthetic data-generating process, mean-function RMSE is 0.01597 for EVE versus 0.01617 for the output model, variance RMSE is 0.00976 versus 0.01122, and variance correlation is 0.99764 versus 0.99663. These oracle quantities are diagnostics on development seeds, not independent test claims.

Table 16. Controlled-noise tracking summaries over ten seeds.

Noise location	Pre-training correlation	Post-training correlation	Absolute center error
Left	-0.464 $[-0.787, -0.141]$	0.747 $[0.653, 0.842]$	0.379 $[0.146, 0.612]$
Center	0.150 $[-0.086, 0.386]$	0.982 $[0.975, 0.989]$	0.225 $[0.128, 0.322]$
Right	0.425 $[0.052, 0.798]$	0.793 $[0.667, 0.918]$	0.261 $[0.045, 0.476]$

The mean tracking slope across seeds is 1.13 with 95% CI $[0.975, 1.29]$. Unit slope lies within the interval, while the point estimate indicates modest overshoot. Together with the condition-wise correlations, this provides controlled evidence that the learned local covariance follows the manipulated data-generating variance pattern. The experiment therefore supplies the semantic-sensitivity evidence level associated with this known synthetic process.

C Supplementary Formalism

This appendix collects supporting formal material for the probabilistic taxonomy, the generic variational-neuron construction, and its Bayesian extensions.

C.1 BNN weight space, function space, and uncertainty

Different weight configurations can encode the same or nearly the same function, and simple weight-space priors can induce complex function-space priors [5,15]. Gaussian processes provide a direct function-space construction and reveal limiting relationships with infinitely wide networks [33,37]. Operationally, BNN uncertainty concerns posterior uncertainty over predictions or functions after conditioning on data.

In heteroscedastic regression,

$$y \mid x, w \sim \mathcal{N}\big(\mu_w(x), \sigma_w^2(x)\big)\,, \tag{26}$$

uncertainty in w and the conditional variance $\sigma_w^2(x)$ play different roles. The law of total variance gives

$$\mathrm{Var}(y_* \mid x_*, \mathcal{D}) = \mathbb{E}_{p(w|\mathcal{D})}[\mathrm{Var}(y_* \mid x_*, w)] \tag{27}$$

$$+ \mathrm{Var}_{p(w|\mathcal{D})}(\mathbb{E}[y_* \mid x_*, w])\,. \tag{28}$$

The first term is commonly associated with expected data uncertainty and the second with model uncertainty, subject to model adequacy and identifiability [14,20,25]. Calibration and robustness remain empirical properties; approximate BNNs can fail under misspecification, and deep ensembles provide an important non-Bayesian reference [21,28,35].

C.2 Generic variational-neuron pseudocode

Algorithm 1 makes explicit the unit computation represented by Eqs. (9)–(12).

C.3 Bayesian extensions of the variational-neuron construction

C.4 When is a variational neuron Bayesian?

Three levels clarify the terminology. The experiments in Section 6 instantiate Level 1, with point-estimated weights and a local reference-regularized latent state. Levels 2 and 3 formalize Bayesian extensions of the architecture and complete the taxonomic comparison across local-latent and weight-level inference regimes.

Algorithm 1 Generic variational-neuron computation and local variational contribution.

Require: h_i, c_i
Obtain μ_i, σ_i from $q_{\phi_i}(z_i \mid h_i, c_i)$
$\epsilon_i \sim \mathcal{N}(0, I)$; $z_i \leftarrow \mu_i + \sigma_i \odot \epsilon_i$
$a_i \leftarrow F_{\eta_i}(h_i, z_i)$
$\mathcal{K}_i \leftarrow \mathrm{KL}(q_{\phi_i}(z_i \mid h_i, c_i) \| p_{\psi_i}(z_i \mid c_i))$
return $(a_i, \mathcal{K}_i)$

Level 1: variationally regularized stochastic architecture. Weights are point-estimated, each unit uses $q_{\phi_i}(z_i \mid h_i)$ and a KL penalty to a reference, and prediction may use samples or means. This defines the variationally regularized stochastic Level 1 regime, distinct from weight-posterior BNN inference.

Level 2: Bayesian inference over local latent variables. A coherent joint density may be written as

$$p_\Theta(y, z_{1:L} \mid x) = p_\Theta(y \mid h_L, z_L) \prod_{\ell=1}^{L} p_{\psi_\ell}(z_\ell \mid h_\ell, c_\ell), \tag{29}$$

with an inference family such as $q_\Phi(z_{1:L} \mid x, y)$ and prediction integrating over local states. Parameters Θ may remain point-estimated.

Level 3: hierarchical Bayesian model with local latent variables (conceptual extension only). Priors are also placed over weights or functions:

$$w \sim p(w), \tag{30}$$
$$z_i \sim p_{\psi_i}(z_i \mid h_i, w, c_i), \tag{31}$$
$$y \sim p(y \mid x, w, \mathbf{z}). \tag{32}$$

Prediction marginalizes both global and local unknowns:

$$p(y_* \mid x_*, \mathcal{D}) = \iint p(y_* \mid x_*, w, z_*) p(z_* \mid x_*, w) p(w \mid \mathcal{D})\, dz_*\, dw. \tag{33}$$

This construction is simultaneously Bayesian, variational when VI is used, stochastic, and distributional at the unit level. The properties remain conceptually distinct.

References

1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion **76**, 243–297 (2021). https://doi.org/10.1016/j.inffus.2021.05.008

2. Achille, A., Soatto, S.: Information dropout: Learning optimal representations through noisy computation. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2897–2905 (2018). https://doi.org/10.1109/TPAMI.2017.2784440
3. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. In: International Conference on Learning Representations (2017)
4. Amini, A., Schwarting, W., Soleimany, A.P., Rus, D.: Deep evidential regression. In: Advances in Neural Information Processing Systems. vol. 33 (2020)
5. Arbel, J., Pitas, K., Vladimirova, M., Fortuin, V.: A primer on Bayesian neural networks: Review and debates. Statistical Science **41**(2), 316–353 (28 May 2026). https://doi.org/10.1214/24-STS969
6. Bellemare, M.G., Dabney, W., Munos, R.: A distributional perspective on reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 449–458. PMLR (2017)
7. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013). https://doi.org/10.48550/arXiv.1308.3432
8. Bishop, C.M.: Mixture density networks. Tech. Rep. NCRG/94/004, Aston University (1994)
9. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
10. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American Statistical Association **112**(518), 859–877 (2017). https://doi.org/10.1080/01621459.2017.1285773
11. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1613–1622. PMLR (2015)
12. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. In: International Conference on Learning Representations (2016)
13. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. In: Advances in Neural Information Processing Systems. vol. 28 (2015)
14. Depeweg, S., Hernández-Lobato, J.M., Doshi-Velez, F., Udluft, S.: Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1184–1193. PMLR (2018)
15. Fortuin, V.: Priors in Bayesian deep learning: A review. International Statistical Review **90**(3), 563–591 (2022). https://doi.org/10.1111/insr.12502
16. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1050–1059. PMLR (2016)
17. Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., et al.: A survey of uncertainty in deep neural networks. Artificial Intelligence Review **56**(Suppl 1), 1513–1589 (2023). https://doi.org/10.1007/s10462-023-10562-9
18. Graves, A.: Practical variational inference for neural networks. In: Advances in Neural Information Processing Systems. vol. 24 (2011)

19. Hernández-Lobato, J.M., Adams, R.P.: Probabilistic backpropagation for scalable learning of Bayesian neural networks. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1861–1869. PMLR (2015)
20. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Machine Learning **110**(3), 457–506 (2021). https://doi.org/10.1007/s10994-021-05946-3
21. Izmailov, P., Nicholson, P., Lotfi, S., Wilson, A.G.: Dangers of Bayesian model averaging under covariate shift. In: Advances in Neural Information Processing Systems. vol. 34 (2021)
22. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79
23. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with Gumbel–Softmax. In: International Conference on Learning Representations (2017)
24. Jospin, L.V., Laga, H., Boussaid, F., Buntine, W., Bennamoun, M.: Hands-on Bayesian neural networks—a tutorial for deep learning users. IEEE Computational Intelligence Magazine **17**(2), 29–48 (2022). https://doi.org/10.1109/MCI.2022.3155327
25. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems. vol. 30 (2017)
26. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. In: Advances in Neural Information Processing Systems. vol. 28 (2015)
27. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations (2014). https://doi.org/10.48550/arXiv.1312.6114
28. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
29. MacKay, D.J.C.: A practical Bayesian framework for backpropagation networks. Neural Computation **4**(3), 448–472 (1992). https://doi.org/10.1162/neco.1992.4.3.448
30. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. In: International Conference on Learning Representations (2017)
31. Maddox, W.J., Garipov, T., Izmailov, P., Vetrov, D., Wilson, A.G.: A simple baseline for Bayesian uncertainty in deep learning. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
32. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: Advances in Neural Information Processing Systems. vol. 31 (2018)
33. Neal, R.M.: Bayesian Learning for Neural Networks, Lecture Notes in Statistics, vol. 118. Springer, New York (1996). https://doi.org/10.1007/978-1-4612-0745-0
34. Oh, S.J., Murphy, K.P., Pan, J., Roth, J., Schroff, F., Gallagher, A.C.: Modeling uncertainty with hedged instance embeddings. In: International Conference on Learning Representations (2019)
35. Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., et al.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In: Advances in Neural Information Processing Systems. vol. 32 (2019)

36. Ranganath, R., Tang, L., Charlin, L., Blei, D.M.: Deep exponential families. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 38, pp. 762–771 (2015)
37. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006)
38. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 1530–1538. PMLR (2015)
39. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 1278–1286. PMLR (2014)
40. Ritter, H., Botev, A., Barber, D.: Online structured Laplace approximations for overcoming catastrophic forgetting. In: Advances in Neural Information Processing Systems. vol. 31 (2018)
41. Ruffenach, Y.: Variational distributional neuron. arXiv preprint arXiv:2602.18250v1 (20 Feb 2026). https://doi.org/10.48550/arXiv.2602.18250
42. Ruffenach, Y.: Exploring the dimensions of a variational neuron. arXiv preprint arXiv:2603.13849v1 (14 Mar 2026). https://doi.org/10.48550/arXiv.2603.13849
43. Sensoy, M., Kaplan, L., Kandemir, M.: Evidential deep learning to quantify classification uncertainty. In: Advances in Neural Information Processing Systems. vol. 31 (2018)
44. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6902–6911 (2019)
45. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. In: Advances in Neural Information Processing Systems. vol. 29 (2016)
46. Tang, C., Salakhutdinov, R.R.: Learning stochastic feedforward neural networks. In: Advances in Neural Information Processing Systems. vol. 26 (2013)
47. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning. pp. 681–688 (2011)
48. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning **8**, 229–256 (1992). https://doi.org/10.1007/BF00992696
49. Wilson, A.G., Izmailov, P.: Bayesian deep learning and a probabilistic perspective of generalization. In: Advances in Neural Information Processing Systems. vol. 33, pp. 4697–4708 (2020)
50. Bayer, J., Osendorfer, C., Urban, S., van der Smagt, P.: Training neural networks with implicit variance. In: Neural Information Processing, LNCS, vol. 8227, pp. 132–139. Springer (2013). https://doi.org/10.1007/978-3-642-42042-9_17
51. Bui, L.M., Tran Huu, T., Dinh, D., Nguyen, T.M., Hoang, T.N.: Revisiting kernel attention with correlated Gaussian process representation. In: Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence. PMLR 244, 450–470 (2024). https://proceedings.mlr.press/v244/bui24a.html
52. Diamzon, J., Venturi, D.: Uncertainty propagation in feed-forward neural network models. Neural Networks **194**, 108178 (Feb 2026). https://doi.org/10.1016/j.neunet.2025.108178

53. Gast, J., Roth, S.: Lightweight probabilistic deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3369–3378 (2018).
54. Hu, J., Yi, X., Li, W., Sun, M., Xie, X.: Fuse it more deeply! A variational Transformer with layer-wise latent variable inference for text generation. In: NAACL-HLT, pp. 697–716 (2022). https://doi.org/10.18653/v1/2022.naacl-main.51
55. Kingma, D.P., Welling, M.: Efficient gradient-based inference through transformations between Bayes nets and neural nets. In: Proceedings of the 31st International Conference on Machine Learning. PMLR 32, 1782–1790 (2014). https://proceedings.mlr.press/v32/kingma14.html
56. Kofnov, A., Kapla, D., Bartocci, E., Bura, E.: Exact upper and lower bounds for the output distribution of neural networks with random inputs. In: Proceedings of the 42nd International Conference on Machine Learning. PMLR 267, 31133–31157 (2025). https://proceedings.mlr.press/v267/kofnov25a.html
57. Luis, C.E., Bottero, A.G., Vinogradska, J., Berkenkamp, F., Peters, J.: Uncertainty representations in state-space layers for deep reinforcement learning under partial observability. Transactions on Machine Learning Research, 1–31 (2025). https://openreview.net/forum?id=rfPns0WJyg
58. Monchot, P., Coquelin, L., Petit, S.J., Marmin, S., Le Pennec, E., Fischer, N.: Input uncertainty propagation through trained neural networks. In: Proceedings of the 40th International Conference on Machine Learning. PMLR 202, 25140–25173 (2023). https://proceedings.mlr.press/v202/monchot23a.html
59. Nguyen, T.T., Choi, J.: Markov information bottleneck to improve information flow in stochastic neural networks. Entropy **21**(10), 976 (2019). https://doi.org/10.3390/e21100976
60. Sankararaman, K.A., Wang, S., Fang, H.: BayesFormer: Transformer with uncertainty estimation. arXiv:2206.00826 (2022). https://doi.org/10.48550/arXiv.2206.00826
61. Shekhovtsov, A., Flach, B., Busta, M.: Feed-forward uncertainty propagation in belief and neural networks. arXiv:1803.10590 (2018). https://doi.org/10.48550/arXiv.1803.10590
62. Wang, H., Shi, X., Yeung, D.Y.: Natural-parameter networks: A class of probabilistic neural networks. In: Advances in Neural Information Processing Systems, vol. 29, pp. 118–126 (2016).
63. Wei, Y., Khardon, R.: Variational inference on the final-layer output of neural networks. Transactions on Machine Learning Research (2024). https://openreview.net/forum?id=mTOzXLmLKr
64. Wright, O., Nakahira, Y., Moura, J.M.F.: An analytic solution to covariance propagation in neural networks. In: Proceedings of the 27th International Conference on Artificial Intelligence and Statistics. PMLR 238, 4087–4095 (2024). https://proceedings.mlr.press/v238/wright24a.html
65. Xue, B., Yu, J., Xu, J., Liu, S., Hu, S., Ye, Z., et al.: Bayesian Transformer language models for speech recognition. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7378–7382 (2021). https://doi.org/10.1109/ICASSP39728.2021.9414046
66. Bahuleyan, H., Mou, L., Vechtomova, O., Poupart, P.: Variational attention for sequence-to-sequence models. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1672–1682. Association for Computational Linguistics, Santa Fe, New Mexico, USA (2018). https://aclanthology.org/C18-1142/

67. Dhuliawala, S.Z., Sachan, M., Allen, C.: Variational classification: A probabilistic generalization of the softmax classifier. Transactions on Machine Learning Research (2024). https://openreview.net/forum?id=EWv9XGOpB3