Learning Distributions Inside a Language Model: Variational Neurons for Measurable Internal Uncertainty and Reliability Monitoring

Yves Ruffenach
Conservatoire National des Arts et Métiers
Strasbourg, France
yves@ruffenach.net
ORCID: 0009-0009-4737-0555
DOI: 10.5281/zenodo.21632472

Abstract

Language models commonly represent hidden computation through deterministic activations, while uncertainty is estimated primarily from their outputs. We introduce Elemental Variational Expanse (EVE), an architecture built from variational distributional neurons that represent selected hidden activations as input-conditioned posterior distributions. Each unit produces a posterior mean and scale, samples a latent activation through reparameterization, and exposes local Kullback–Leibler activity together with activation and disagreement diagnostics during the forward pass. This design makes probabilistic structure directly observable within the computation and connects neuron-level dynamics with predictive behavior, calibration, and robustness. We evaluate EVE in controlled next-token prediction against deterministic, Monte Carlo Dropout, and ensemble references, supported by ablations and non-stationarity analyses that examine stochastic sampling, posterior scale, regularization, and internal–external relationships. Across these evaluations, EVE improves predictive quality, provides informative internal diagnostics, and supports reliability-oriented monitoring and readout strategies. Variational distributional neurons therefore offer a general computational primitive for uncertainty-aware language models and, more broadly, for neural systems designed to preserve and use multiple plausible internal states.

Keywords: variational neurons; distributional activations; language modeling; internal probabilistic activity; uncertainty quantification; calibration; tail risk; distribution shift; reliability monitoring

1 Introduction

A language model encounters several forms of indeterminacy. The next token may admit multiple plausible continuations; the training data may support several interpretations; the parameter configuration may represent a broad family of hypotheses; or the input may depart from familiar training conditions. These situations are commonly grouped under the term *uncertainty*. They arise at different levels of the system and motivate distinct responses.

Most contemporary neural networks nevertheless perform their internal computation with point-valued activations. A conventional hidden unit maps an input representation x to one value,

$$h = f(Wx + b). \tag{1}$$

Ambiguity is commonly assessed from output probabilities, repeated dropout passes, a posterior over weights, or disagreement among separately trained models (Blundell et al., 2015; Gal and

Ghahramani, 2016; Lakshminarayanan et al., 2017; Guo et al., 2017). These approaches motivate a complementary architectural question:

> *Can probability be represented directly inside the elementary hidden computation and remain observable throughout prediction?*

We investigate this question through a *variational neuron*. Instead of producing only one activation, the unit predicts the parameters of an input-conditioned distribution,

$$q_\theta(z \mid x) = \mathcal{N}\Big(\mu_\theta(x), \sigma_\theta^2(x)\Big), \qquad z = \mu_\theta(x) + \sigma_\theta(x)\epsilon, \quad \epsilon \sim \mathcal{N}(0,1). \tag{2}$$

The sampled value z is propagated as the activation. The mean $\mu_\theta(x)$ describes the central activation proposed by the unit, while $\sigma_\theta(x)$ controls the dispersion of the possible activations around that mean. A Kullback–Leibler term constrains this local posterior relative to a prior. Throughout the paper, *posterior* denotes this amortized, input-conditioned variational distribution over the unit state.

This construction changes *what is probabilistic*. In a Bayesian neural network, the random object is typically a weight or a collection of weights. In Monte Carlo Dropout, it is a mask applied to a deterministic network. In a deep ensemble, it is the selected trained model. In a variational autoencoder or information bottleneck, it is usually a vector representation associated with an example or a layer. In the construction studied here, the random object is the activation state of each selected computational unit for the current input.

A concrete intuition. Suppose an ordinary neuron receives a context and returns 0.8. The downstream network sees only that value. A variational neuron may instead return a posterior with mean 0.8 and scale 0.1 for one context and mean 0.8 and scale 1.2 for another. The central activation is identical, yet the two computations have different internal dispersion. Repeated forward passes can therefore produce a family of activations and predictions. The architecture also exposes quantities such as posterior scale, local KL activity, posterior mean energy, and sample disagreement while the prediction is being formed.

Posterior scale is interpreted first as a model-internal dispersion parameter. Its role as an uncertainty diagnostic is evaluated empirically through sensitivity to difficult or shifted inputs, association with predictive risk, comparison with output confidence scores, and integration into operational policies. This yields three complementary levels: *probabilistic activity*, *uncertainty diagnostics*, and *actionable uncertainty*.

Elemental Variational Expanse (EVE) is a controlled language-model implementation of this idea. Frozen GPT-2 token embeddings feed a trainable architecture whose central bank contains 1,024 input-conditioned Gaussian units. The model is trained for next-token prediction on WritingPrompts-Filtered. Its stochastic activations are sampled during training and evaluation, and the experiment records both external performance and internal posterior statistics.

The study addresses four progressively stronger questions:

1. **Representation:** can local activation distributions be learned and measured inside the forward computation?
2. **Mechanism:** do sampling, learned posterior scale, and KL regularization make distinguishable contributions?
3. **Prediction and monitoring:** does the resulting model improve predictive metrics, and do its internal statistics change with external risk?
4. **Operational use:** how can internal statistics be organized into an explicit rule for abstention, resampling, deferral, or fallback?

The experiments establish the representation, mechanism, prediction, and monitoring levels. Across five shared random seeds, raw EVE improves Monte Carlo negative log-likelihood, perplexity, accuracy, and 5%-tail CVaR-NLL relative to the raw deterministic and Monte Carlo Dropout baselines. Mean-only, no-KL, and fixed-scale ablations show that stochastic sampling, learned dispersion, and posterior regularization contribute through distinct mechanisms. Under several controlled shifts, changes in internal KL and posterior mean energy correlate with external risk. Validation-selected readouts characterize global output-sample policies, and the formal internal-score policy specifies the input-dependent extension toward abstention, resampling, fallback, retrieval, or human review.

The empirical analysis centers on unit-level probabilistic representation, mechanism analysis, predictive performance, and reliability monitoring in a controlled language-model setting.

The main contributions are:

1. **A unit-level probabilistic formulation.** We define a Gaussian activation posterior attached to a local hidden computation and specify the quantities observable during a forward pass.
2. **A complete architectural account.** We report the placement and number of variational units, the controller and regulator, sampling procedure, training losses, parameter counts, and heuristic compute.
3. **A matched-seed predictive study.** We compare raw EVE with deterministic and Monte Carlo Dropout models across five common seeds and retain a five-member deterministic ensemble as an additional aggregation reference.
4. **Mechanism-oriented ablations.** Mean-only inference, no-KL training, and fixed posterior scale separate the effects of stochastic sampling, learned dispersion, and regularization.
5. **Internal monitoring under shift.** We examine local KL, posterior scale, posterior mean energy, mutual information, and prediction instability under multiple non-stationarity conditions.
6. **An operational formulation for actionability.** We distinguish global output-sample readouts from an input-dependent internal-signal policy and provide an explicit decision rule based on internal statistics.

2 Related work and positioning

The idea of a probabilistic computational unit lies at the intersection of several literatures that use similar mathematics for different random objects. The central distinction concerns *where the probability distribution lives, what it represents, and how it is used.*

2.1 From point-valued neurons to stochastic neural computation

Early formal neurons were usually described as thresholded or point-valued computational elements (McCulloch and Pitts, 1943; Rosenblatt, 1958). Stochastic units nevertheless have a long history. Boltzmann machines use stochastic binary units in an energy-based model, with learning driven by differences between data-conditioned and free-running statistics (Ackley et al., 1985). Sigmoid belief networks and the wake–sleep algorithm use directed layers of stochastic latent units to build generative models (Hinton et al., 1995). Continuous sigmoidal belief networks inject Gaussian noise before a nonlinear activation, allowing behavior ranging from nearly deterministic to continuously stochastic (Frey, 1996).

These models establish that randomness can be part of the hidden computation itself. Their main purposes include generative modeling, energy-based inference, and multimodal conditional prediction. EVE studies a reparameterized Gaussian activation posterior inside a discriminatively

trained next-token model and treats its local posterior statistics as observable diagnostics.

2.2 Stochastic hidden units and noisy activations

Stochastic feedforward neural networks introduced mixtures of deterministic and stochastic hidden variables to represent multimodal conditional distributions (Tang and Salakhutdinov, 2013). Other work constructed stochastic units from monotonic activation functions through Gaussian approximations (Ravanbakhsh et al., 2016), or studied gradient estimators for deep stochastic binary networks (Shekhovtsov et al., 2020). This literature is important because it separates *stochastic hidden computation* from Bayesian uncertainty over parameters.

The closest conceptual overlap is the replacement of a point activation by a random hidden state. The difference lies in the parameterization and the scientific question. EVE uses an amortized, input-conditioned Gaussian posterior with explicit local KL and reconstruction terms, records posterior statistics per unit, and evaluates whether those statistics track language-model reliability.

2.3 Variational inference, stochastic backpropagation, and local reparameterization

Modern differentiable latent-variable models became practical through stochastic variational inference and pathwise gradient estimators. Variational autoencoders parameterize an approximate posterior with a neural encoder and optimize an evidence lower bound through reparameterized samples (Kingma and Welling, 2014). Stochastic backpropagation provides the corresponding general treatment for deep generative models (Rezende et al., 2014), while neural variational inference extended scalable amortized inference to belief networks with discrete stochastic variables (Mnih and Gregor, 2014).

The local reparameterization trick converts uncertainty in global weights into independent activation noise for each example, reducing gradient variance and connecting Gaussian dropout with variational inference (Kingma et al., 2015). This is mathematically close to sampling activations, with a distinct semantic interpretation. Under local reparameterization, the activation distribution is induced by a posterior over weights; it is primarily a computational device for estimating a Bayesian objective. In EVE, the activation posterior is directly parameterized as the modeled state of the unit and its parameters are retained as first-class internal observables.

2.4 Bayesian neural networks, dropout, and ensembles

Bayesian neural networks place distributions over parameters and integrate or approximate them at prediction time. Bayes by Backprop learns factorized weight posteriors with a variational objective (Blundell et al., 2015); matrix-Gaussian and structured posteriors model correlations among weights (Louizos and Welling, 2016; Sun et al., 2017). Variational Dropout and related sparsification methods learn multiplicative noise or dropout rates under Bayesian interpretations (Kingma et al., 2015; Molchanov et al., 2017). Subsequent analyses refine the modeling assumptions and variational formulations underlying dropout (Hron et al., 2018; Nalisnick et al., 2019).

Monte Carlo Dropout samples masks from one trained network (Gal and Ghahramani, 2016). Deep ensembles sample neither weights nor activations from an explicit posterior; they aggregate independently optimized deterministic models and remain a strong practical uncertainty baseline (Lakshminarayanan et al., 2017). All of these approaches can induce a distribution over hidden activations. The distinction is that their primary stochastic object is a weight, mask, or model index. EVE directly represents the current unit activation as an input-conditioned posterior.

EVE provides a distinct *computational granularity*: uncertainty-related quantities are available locally through the activation posterior, alongside parameter-level and model-level approaches.

2.5 Function-space uncertainty, deep Gaussian processes, and Gaussian Process Neurons

Gaussian processes place probability distributions over functions. Deep Gaussian processes compose probabilistic mappings and perform approximate variational marginalization across layers (Damianou and Lawrence, 2013). Infinite-width neural networks can also converge to Gaussian-process limits, connecting parameter priors to function-space uncertainty (Lee et al., 2018).

Gaussian Process Neurons are especially relevant. They place a Gaussian-process prior over the activation function associated with an individual neuron and infer a posterior distribution over that function (Urban et al., 2017). Their stochastic object is therefore the *function implemented by the neuron*. In EVE, the maps producing $\mu(x)$ and $\sigma(x)$ are deterministic, while the stochastic object is the *activation state generated for the current input*. Function uncertainty and state uncertainty provide related and complementary ways of constructing a probabilistic unit.

2.6 Variational information bottlenecks and distributional representation layers

The Deep Variational Information Bottleneck (VIB) introduces a stochastic representation Z that compresses the input while retaining information useful for prediction (Alemi et al., 2017). Later work used VIB objectives to prune neurons and compress networks (Dai et al., 2018). The recently proposed Cell Variational Information Bottleneck Network stacks cells that each predict Gaussian means and standard deviations and generate uncertain feature maps through reparameterization (Zhai, 2024).

This is one of the closest architectural families to EVE. Both VIB-style layers and EVE learn input-conditioned Gaussian hidden representations and regularize them with KL terms. The intended granularity and evaluation differ. VIB typically treats the stochastic code as a layer- or representation-level information bottleneck, with compression or robustness as the central objective. CellVIB distributes that principle across convolutional cells. EVE resolves the code into a bank of explicitly monitored local units in a language-model core, adds unit-level activity control and local reconstruction, and studies posterior statistics as reliability-monitoring signals. The distinction is thus not the mere presence of μ, σ, reparameterization, or KL; it is the unit-level instrumentation, language-model setting, and diagnostic objective.

2.7 Structured stochastic neural networks and learned hidden noise

Several approaches learn structured noise directly inside a network. Kronecker-flow stochastic neural networks parameterize high-dimensional correlated noise through expressive variational transformations (Huang et al., 2020). Learned anisotropic hidden noise has also been used to optimize robustness-related bounds (Eustratiadis et al., 2021). Recent analyses show that the training objective and number of Monte Carlo samples can materially change the predictive variance and out-of-distribution behavior of stochastic neural networks (Däubener et al., 2025).

These results distinguish predictive gain from semantic interpretation of learned variance. Noise can provide regularization, support optimization, change effective capacity, or alter robustness. The ablations and capacity accounting separate these mechanisms and place the posterior statistics in their architectural context.

2.8 Predictive uncertainty, calibration, and distribution shift

Predictive uncertainty methods are usually evaluated at the output level. Temperature scaling can improve calibration without changing the learned representation (Guo et al., 2017). Deep ensembles often provide strong predictive and calibration performance (Lakshminarayanan et al., 2017). Prior Networks parameterize a distribution over predictive distributions to distinguish data, model, and distributional uncertainty (Malinin and Gales, 2018). Under dataset shift, the quality of uncertainty estimates varies substantially across methods and corruption types (Ovadia et al., 2019). Simple softmax-based scores remain important controls for misclassification and out-of-distribution detection (Hendrycks and Gimpel, 2017).

This literature provides reference criteria for interpreting EVE's internal posterior statistics, including output confidence, entropy, Monte Carlo variance, and ensemble disagreement. The present study focuses on internal observability, predictive comparison, mechanism ablations, and aggregate response under distribution shift.

2.9 Uncertainty in language models

Language generation adds a further difficulty: several token sequences can express the same meaning. Token entropy can therefore treat paraphrases as distinct outcomes even when they are semantically equivalent. Semantic entropy addresses this by grouping generations according to meaning before estimating uncertainty (Kuhn et al., 2023). Instance-level analyses have also compared model sampling variability with human production variability (Giulianelli et al., 2023), while rank-calibration evaluates whether larger uncertainty scores correspond monotonically to lower generation quality (Huang et al., 2024).

Other work trains language models to verbalize calibrated probabilities (Kapoor et al., 2024) or learns confidence predictors from internal activations and uses them to guide decoding (Liu et al., 2024). These studies connect an uncertainty signal to a concrete prediction or decoding policy. EVE contributes the complementary architectural layer: probabilistic states are built into hidden units and monitored during next-token prediction. Semantic uncertainty, factual question answering, open-ended generation, and activation-conditioned guided decoding provide natural application domains for this mechanism.

2.10 Interpretability, internal monitoring, and causal intervention

Feature attribution and local surrogate methods explain a prediction in terms of input features (Ribeiro et al., 2016; Lundberg and Lee, 2017; Sundararajan et al., 2017). Network dissection associates hidden units with human-interpretable concepts (Bau et al., 2017). Work on transformer circuits and feed-forward layers studies the organization of internal computations (Elhage et al., 2021; Geva et al., 2021), while causal tracing and model editing intervene on internal states to test their role in factual behavior (Meng et al., 2022).

EVE contributes unit-level instrumentation through observable posterior statistics. Localization, semantic attribution, controlled interventions, and predictable behavioral changes can extend this instrumentation into a mechanistic account of individual units.

2.11 Selective prediction and operational actionability

Selective prediction turns uncertainty into an action by allowing a model to abstain on difficult examples. Risk–coverage evaluation measures the error among accepted examples as coverage changes (Geifman and El-Yaniv, 2017); SelectiveNet integrates the reject option into end-to-end

learning (Geifman and El-Yaniv, 2019). This framework provides a precise standard for the phrase *actionable uncertainty*: a score should change an operational decision and improve a predefined risk or cost function against appropriate controls.

The validation-selected temperatures and output-sample readouts provide an initial operational layer. A direct internal-signal policy uses per-input EVE statistics to trigger abstention, additional sampling, fallback, retrieval, or human review, with output entropy, maximum probability, semantic entropy, MC variance, random rankings, and constant policies as comparison scores.

2.12 Where EVE sits

Table 1 summarizes the main distinction across neighboring approaches.

Family	Primary random object	Typical purpose or observable	Relation to EVE
Boltzmann machines / belief networks	binary or continuous hidden variables	generative modeling and latent inference	establishes stochastic neurons with generative objectives; EVE uses an amortized Gaussian activation posterior for LM reliability monitoring
Bayesian neural networks	weights or weight matrices	parameter posterior and predictive integration	induces random activations indirectly; EVE parameterizes the activation posterior directly
MC Dropout	Bernoulli masks / multiplicative noise	approximate uncertainty and regularization	samples a deterministic network under masks; EVE parameterizes an explicit local Gaussian posterior
Deep ensembles	trained model index	predictive disagreement	strong output-level reference; each member remains internally deterministic
VAE / Deep VIB	example- or layer-level latent code	generation, compression, or robust representation learning	shares reparameterization and KL; EVE emphasizes local units and internal monitoring
CellVIB	stochastic feature-map cells	distributed information bottlenecks and robustness	closest layerwise architectural relative; different domain, unit instrumentation, controller, and diagnostic objective
Deep GPs / GP neurons	functions or activation functions	function-space uncertainty	models probabilistic functions; EVE models the current activation state
Prior Networks	distribution over predictive distributions	data and distributional uncertainty	models an output-distribution hierarchy; EVE models hidden-unit posteriors
Activation-based LM confidence	deterministic hidden activations used by a probe	confidence calibration or guided decoding	uses existing activations as features; EVE makes the activations themselves stochastic and distributional
EVE	input-conditioned activation state of each selected unit	measurable local posterior activity, stochastic prediction, and reliability monitoring	unit-level Gaussian state with explicit local observables

Table 1: Positioning by the location and meaning of the modeled probability distribution. Categories can overlap, but their primary random objects and intended uses differ.

The closest precedents come from three directions: stochastic hidden-unit networks, variational information-bottleneck layers, and Gaussian Process Neurons. EVE builds on this lineage by combining an explicitly unit-indexed activation posterior, local diagnostics and activity control, a language-modeling test bed, mechanism ablations, and reliability-oriented stress monitoring in one controlled study.

3 Interpretation of internal quantities

3.1 Probabilistic activity and uncertainty diagnostics

For a local posterior $q(z \mid x) = \mathcal{N}(\mu(x), \sigma^2(x))$, posterior scale and KL divergence describe the local distribution. Mutual information and top-1 instability quantify sample disagreement. The quantity

μ^2 measures *posterior mean energy*, or activation magnitude, and records the computational activity of the variational pathway.

3.2 Empirical findings

The experiment supports the following statements:

- a language-model unit can carry a learned input-conditioned activation posterior;
- posterior statistics can be recorded during the forward pass;
- stochastic sampling, learned scale, and KL regularization affect prediction and internal regularity in different ways;
- internal statistics respond systematically under several controlled shifts;
- raw EVE improves the reported predictive metrics over the raw deterministic and Monte Carlo Dropout baselines in this controlled setting.

4 Method

4.1 Local variational activation

Let h be the representation entering a variational unit. The unit predicts a posterior mean and log-scale,

$$\mu = f_\mu(h), \tag{3}$$

$$\log \sigma^2 = f_\sigma(h), \tag{4}$$

and defines

$$q_\theta(z \mid h) = \mathcal{N}\Big(\mu(h), \operatorname{diag}(\sigma^2(h))\Big). \tag{5}$$

A latent activation is sampled by reparameterization,

$$z = \mu(h) + \sigma(h) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{6}$$

The prior is standard normal. For one local latent state,

$$\mathrm{KL}(q_\theta(z \mid h) \| p(z)) = \frac{1}{2} \sum_j \Big(\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1\Big). \tag{7}$$

The task objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \beta\, \mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{rec}} \mathcal{L}_{\mathrm{local-rec}} + \lambda_{\mathrm{band}} \mathcal{L}_{\mathrm{band}}. \tag{8}$$

The exported configuration uses a base KL coefficient of 10^{-4}, local reconstruction weight 0.03 during the first phase and 0.022 from epoch 3, and a soft activity-band penalty. The posterior scale is capped at 7.0 initially and 5.8 from epoch 3.

4.2 Internal observables

For U measured units and input x, we record

$$A_{\mathrm{KL}}(x) = \frac{1}{U} \sum_{u=1}^{U} \mathrm{KL}_u(x), \tag{9}$$

$$A_{\sigma}(x) = \frac{1}{U} \sum_{u=1}^{U} \sigma_u(x), \tag{10}$$

$$A_{\mu^2}(x) = \frac{1}{U} \sum_{u=1}^{U} \mu_u(x)^2. \tag{11}$$

A_{KL} and A_{σ} are posterior diagnostics. A_{μ^2} is posterior mean energy and records the magnitude of the mean activation pathway.

Across S stochastic predictive passes, we additionally compute mutual information and top-1 instability. These measure disagreement induced by latent sampling. Their interpretation depends on context: larger values can reflect predictive diversity, sensitivity, or changing latent participation.

4.3 Local reconstruction and regulator

Each variational unit includes a one-dimensional local decoder that reconstructs a detached projection of the incoming representation. The reconstruction term encourages sampled latent states to retain input-dependent information.

The terms *homeostatic controller* and *neural regulator* refer to the activity-control mechanism rather than to an additional predictive network. The controller tracks an exponential moving average of posterior mean energy per unit. The configured target is 0.10, with a maintained band from 0.08 to 0.145. After a 200-step warm-up, bounded row-wise rescaling of the posterior-mean projection is applied every 10 steps with update strength 0.22 and maximum multiplicative step 1.35. The band penalty maintains posterior activity within the configured operating range. These controls are specified explicitly because they contribute to optimization and form part of the complete variational-neuron configuration.

4.4 Architecture and placement

The input tokens are mapped through the frozen GPT-2 embedding matrix. The trainable EVE pathway contains a bank of 1,024 variational units with latent dimension $k = 1$ and per-unit, input-dependent posterior scale. The resulting representation is processed by three densely connected variational multilayer-perceptron stages of width 1,024, a transformer-style contextual block with 12 attention heads and positional embeddings, and a gated decoder of width 1,024 and depth 1. Dropout is 0.05 in the variational MLP, attention, residual, and decoder pathways. Thus, the variational units are not sparsely inserted at an unspecified location: the entire 1,024-unit hidden bank at the model core is distributional in this experiment.

The deterministic baseline uses the same tokenization, frozen embedding matrix, data windows, training epochs, and evaluation protocol. Its hidden computation is point-valued and its trainable capacity is smaller, yielding a protocol-aligned comparison with explicit capacity accounting.

4.5 Prediction and Monte Carlo aggregation

Let $S = 4$ denote the number of stochastic predictions used for the reported metrics, let ℓ_s be the logits from sample s, and define

$$p_s^{(T)} = \mathrm{softmax}(\ell_s / T). \tag{12}$$

The raw EVE predictive distribution is the mean of the sample probabilities at $T = 1$,

$$\bar{p} = \frac{1}{S} \sum_{s=1}^{S} p_s^{(1)}. \tag{13}$$

4.6 Validation-selected readouts

The secondary readout analysis uses the same frozen sample bank and searches five aggregation rules:

1. mean probabilities: $S^{-1} \sum_s p_s^{(T)}$;
2. softmax of mean logits: $\mathrm{softmax}(S^{-1} \sum_s \ell_s / T)$;
3. coordinate-wise median probabilities followed by renormalization;
4. entropy-weighted probabilities;
5. confidence-weighted probabilities.

For entropy weighting,

$$H_s = -\sum_v p_{s,v}^{(T)} \log p_{s,v}^{(T)}, \tag{14}$$

$$w_s^{(H)} = \frac{\exp(-H_s / \tau_w)}{\sum_r \exp(-H_r / \tau_w)}. \tag{15}$$

For confidence weighting, $c_s = \max_v p_{s,v}^{(T)}$ and

$$w_s^{(C)} = \frac{\exp(c_s / \tau_w)}{\sum_r \exp(c_r / \tau_w)}, \tag{16}$$

with $\tau_w = 0.25$. Weighted predictions use $\sum_s w_s p_s^{(T)}$.

Temperature is searched over

$$T \in \{0.8, 0.9, 1.0, 1.1, 1.2, 1.35, 1.5, 1.75, 2.0, 2.25, 2.5, 3.0\}. \tag{17}$$

Selection uses validation data only. The scores are

$$S_{\mathrm{ECE}} = \mathrm{ECE} + 0.01\mathrm{NLL} + 0.001 \max(0, \text{CVaR-NLL} - \mathrm{NLL}), \tag{18}$$

$$S_{\mathrm{CVaR}} = \text{CVaR-NLL} + 0.05\mathrm{NLL} + 0.25\mathrm{ECE}, \tag{19}$$

$$S_{\mathrm{bal}} = \mathrm{NLL} + 2\mathrm{ECE} + 0.05 \max(0, \text{CVaR-NLL} - \mathrm{NLL}). \tag{20}$$

The selected policies are $T = 1.1$ with mean probabilities for the ECE criterion, $T = 3.0$ with median probabilities for the CVaR criterion, and $T = 1.2$ with confidence weighting for the balanced criterion.

These readouts use output probabilities and sample-level entropy or confidence, demonstrating the flexibility of a stochastic posterior bank. The input-dependent policy below adds internal KL, posterior scale, and mutual information as direct decision variables.

4.7 Input-dependent internal-signal policy

An input-dependent action policy can be formulated from the internal score

$$u(x) = aA_{\mathrm{KL}}(x) + bA_{\sigma}(x) + c\mathrm{MI}(x) \tag{21}$$

and choose an action such as acceptance, abstention, additional sampling, or fallback:

$$\pi_{\tau}(x) = \begin{cases} \text{standard prediction}, & u(x) \leq \tau, \\ \text{abstain or escalate}, & u(x) > \tau. \end{cases} \tag{22}$$

The threshold τ and coefficients can be selected on validation data to define an operating rule. Output entropy, maximum probability, and Monte Carlo variance provide natural reference scores for the same decision setting. This formulation translates internal statistics into an explicit operational rule without changing the predictive model.

5 Experimental setup

5.1 Data and task

We use a fixed controlled subset of `RLAIF/WritingPrompts-Filtered` with GPT-2 tokenization and the frozen GPT-2 embedding matrix (Fan et al., 2018; Radford et al., 2019). The subset is partitioned into fixed training, validation, and test windows using a 70/15/15 protocol. Context length is 24, target stride is 2, and all models are trained for four epochs with batch size 48. The same preprocessing, partitioning, and window-generation protocol is applied consistently across all models, seeds, and ablations.

GPT-2 contributes the frozen embedding matrix, while the predictive network is trained specifically for this controlled experiment. The reported losses therefore characterize this benchmark configuration.

5.2 Compared systems

- DET: deterministic hidden computation, five seeds.
- MC Dropout: each deterministic checkpoint is evaluated with dropout active over four stochastic passes, five seeds.
- DET Ensemble: probability average of five independently trained deterministic members; reported as a strong aggregation reference.
- Raw EVE: mean probabilities over four latent samples, five seeds.
- Readout variants: validation-selected transformations of one reference EVE sample bank; secondary diagnostic analysis only.

5.3 Training

Training uses mixed precision, weight decay 10^{-4}, gradient clipping at norm 1.0, learning rate 1.5×10^{-4} for EVE, and 2×10^{-4} for DET. EVE uses four latent samples per optimization step. Metric evaluation uses four stochastic passes under a fixed compute-bounded evaluation protocol applied identically across models, seeds, ablations, and reported operating conditions. Seeds are $\{0, 1, 2, 3, 4\}$.

5.4 Capacity and compute

The frozen embedding matrix contributes 38,597,376 non-trainable parameters to each model. DET and MC Dropout contain 20,715,265 trainable parameters; EVE contains 134,852,868 trainable parameters. EVE therefore has approximately 6.51× as many trainable parameters as DET and 2.92× as many total parameters when frozen embeddings are included.

A conservative accounting estimate gives approximately 1.83×10^{15} training-plus-evaluation FLOPs for one DET run and 4.32×10^{16} for one EVE run under the reported sampling protocol, a ratio of approximately 23.6×. These accounting estimates provide capacity and compute context for the complete-configuration comparison.

Model	Trainable params	Frozen params	Predictive passes	Role
DET	20,715,265	38,597,376	1	raw deterministic baseline
MC Dropout	20,715,265	38,597,376	4	stochastic mask baseline
EVE	134,852,868	38,597,376	4	local activation posterior
DET Ensemble	20,715,265/member	38,597,376/member	5	aggregation reference

Table 2: Model capacity and predictive-pass profile for the protocol-aligned comparison.

5.5 Metrics and the CE–NLL distinction

Cross-entropy (CE) is the teacher-forced optimization and checkpoint-selection loss produced by the model's training/evaluation path. Monte Carlo NLL is computed from the final predictive distribution after aggregating stochastic samples. For a deterministic single-pass model the quantities coincide conceptually, whereas for a stochastic model CE from an individual or designated forward path need not equal the NLL of the aggregated predictive distribution. To avoid redundant or ambiguous comparisons, the primary table reports aggregated NLL and defines perplexity as exp(NLL).

We additionally report next-token accuracy, 15-bin ECE, mutual information, top-1 instability, and CVaR-NLL over the 5% highest-loss evaluated examples. Internal EVE measurements are local KL, posterior scale, and posterior mean energy.

5.6 Distribution-shift protocol

Stress conditions include context-token corruption rates 0.05, 0.15, and 0.30; rare- and common-target slices; short- and long-context conditions; ordered temporal bins; and post-shift recalibration. Together they sample several controlled forms of non-stationarity for diagnostic analysis.

6 Results

6.1 Primary five-seed comparison

Model	Seeds	MC-NLL ↓	PPL ↓	Accuracy ↑	ECE ↓	CVaR-NLL ↓
DET	5	5.620 ± 0.022	275.845 ± 6.015	0.1482 ± 0.0014	0.0426 ± 0.0068	13.200 ± 0.299
MC Dropout	5	5.621 ± 0.021	276.161 ± 5.894	0.1535 ± 0.0100	0.046 ± 0.011	13.213 ± 0.290
DET Ensemble	5 members	5.455	233.81	0.1632	0.0466	12.761
EVE raw	5	$\mathbf{5.359 \pm 0.039}$	$\mathbf{212.743 \pm 8.288}$	$\mathbf{0.1659 \pm 0.0017}$	$\mathbf{0.0418 \pm 0.0087}$	$\mathbf{12.713 \pm 0.212}$

Table 3: Primary predictive comparison. Single-model rows report mean ± sample standard deviation over five shared seeds. The five-member deterministic ensemble is included as an aggregation reference.

Across the shared seeds, raw EVE reduces mean MC-NLL by approximately 4.6% relative to DET, increases accuracy by approximately 11.9%, and reduces mean CVaR-NLL by approximately 3.7%. ECE is comparable. The deterministic ensemble is stronger than a single deterministic model and remains a competitive reference, while raw EVE achieves the leading reported NLL, accuracy, and CVaR values.

These differences characterize the predictive performance of the complete EVE configuration under the shared data, seed, and evaluation protocol. Table 2 provides the corresponding parameter and predictive-pass context, while the ablations separate sampling, scale learning, and KL regularization within the evaluated architecture.

6.2 Internal probabilistic activity

Model	MI	Top-1 instability	Internal KL	Posterior scale
DET	0.000	0.000	–	–
MC Dropout	0.007 ± 0.003	–	–	–
DET Ensemble	0.153	0.292	–	–
EVE raw	0.190 ± 0.011	0.324 ± 0.012	0.280 ± 0.017	0.872 ± 0.005

Table 4: Predictive disagreement and EVE-specific posterior observables. Dashes indicate quantities not reported for the corresponding reference.

The internal quantities establish unit-level observability. A high-KL unit is farther from the standard-normal reference prior, while posterior scale describes local dispersion and sample disagreement records predictive variation across stochastic passes. Together these measurements separate local posterior activity from output-level disagreement.

6.3 Completed mechanism ablations

The ablation study covers seeds 0–4 for raw EVE, mean-only evaluation, no-KL training, and fixed-scale training. Mean-only is an evaluation-time transformation of trained raw checkpoints; no-KL and fixed-scale are separately trained variants.

Variant	Seeds	NLL ↓	Acc. ↑	ECE ↓	CVaR ↓	MI	Top-1 instability	Internal KL	Scale
Raw EVE	5	5.359 ± 0.039	0.1659 ± 0.0017	0.042 ± 0.009	12.713 ± 0.212	0.190 ± 0.011	0.324 ± 0.012	0.280 ± 0.017	0.872 ± 0.005
Mean-only eval	5	7.282 ± 0.161	0.1458 ± 0.0121	0.332 ± 0.030	22.103 ± 0.618	0.000	0.000	0.256	0.868
No-KL	5	5.299 ± 0.138	0.1733 ± 0.0062	0.047 ± 0.007	12.837 ± 0.304	0.053	0.156	2053.880	1.216
Fixed-scale	5	5.412 ± 0.046	0.1611 ± 0.0043	0.042 ± 0.009	12.582 ± 0.139	0.179	0.326	0.700	0.871

Table 5: Completed five-seed mechanism ablations. Mean-only removes stochastic sampling at evaluation; no-KL removes the local prior penalty; fixed-scale freezes posterior scale parameters.

Sampling. Replacing z with μ at evaluation increases NLL, calibration error, and tail risk while reducing sample disagreement to zero. Stochastic latent sampling therefore contributes materially to the learned predictive computation.

Learned scale. The fixed-scale variant remains competitive and achieves the lowest mean CVaR in the ablation table, with a higher NLL than raw EVE. Learned input-dependent scale therefore contributes to the observed posterior profile and predictive trade-off alongside the other architectural components.

KL regularization. Removing KL improves mean short-horizon NLL and accuracy while producing an internal KL value several orders of magnitude larger than raw EVE and a substantially larger scale. KL therefore acts as a structural constraint that maintains the local posterior within a controlled regime, separating predictive fit from posterior geometry.

Together, these variants isolate the roles of stochastic sampling, learned scale, and KL regularization within the evaluated EVE configuration.

6.4 Validation-selected readout analysis

Model/readout	NLL ↓	Acc. ↑	ECE ↓	CVaR ↓
DET	5.592	0.1469	0.046	12.889
MC Dropout	5.595	0.1476	0.048	12.915
EVE raw	5.392	0.1657	0.055	12.836
EVE-ECE	**5.360**	0.1649	**0.044**	12.234
EVE-balanced	5.410	0.1649	0.046	12.237
EVE-tail-risk	7.600	0.1528	0.149	**10.493**
DET Ensemble	5.455	0.1632	0.047	12.761

Table 6: Reference-run validation-selected readout analysis, reported separately from the primary five-seed comparison.

The ECE and balanced policies use distinct transformations: the ECE policy uses $T = 1.1$ with mean probabilities, while the balanced policy uses $T = 1.2$ with confidence-weighted samples. The tail-risk policy uses $T = 3.0$ with median probabilities and attains the lowest reported CVaR in this readout analysis, alongside a deliberate average-likelihood and calibration trade-off. These operating points characterize complementary uses of the same stochastic sample bank.

6.5 Robustness under non-stationarity

Scenario	DET NLL	EVE raw NLL	EVE bal. NLL	DET CVaR	EVE raw CVaR	EVE bal. CVaR
Base test	5.4782	5.2891	5.3140	12.6848	12.2045	11.6761
Covariate 0.05	5.7035	5.5454	5.5583	13.2150	13.0727	12.5267
Covariate 0.15	6.3089	6.2303	6.1809	16.0280	16.0428	15.0554
Covariate 0.30	8.1670	8.0541	7.8379	18.5404	18.5066	17.2270
Rare targets	10.3754	10.0332	9.7323	15.6192	15.8392	14.8581
Common targets	2.0121	2.2899	2.5162	4.3460	4.9992	4.8995

Table 7: Representative multi-axis stress summary. EVE is strongest on the base, covariate, and rare-target conditions, while DET is strongest on the common-target slice.

Under increasing context corruption, external risk and several internal measurements rise together. The highest corruption level produces the largest NLL and CVaR values. The rare-target slice favors the reported EVE variants, while the common-target slice favors DET, revealing complementary strengths across shift families.

The correlations provide a descriptive condition-level view of how internal activity co-varies with external risk across the predefined shift families. This view complements the predictive tables by showing that the internal measurements remain responsive as the evaluation conditions change.

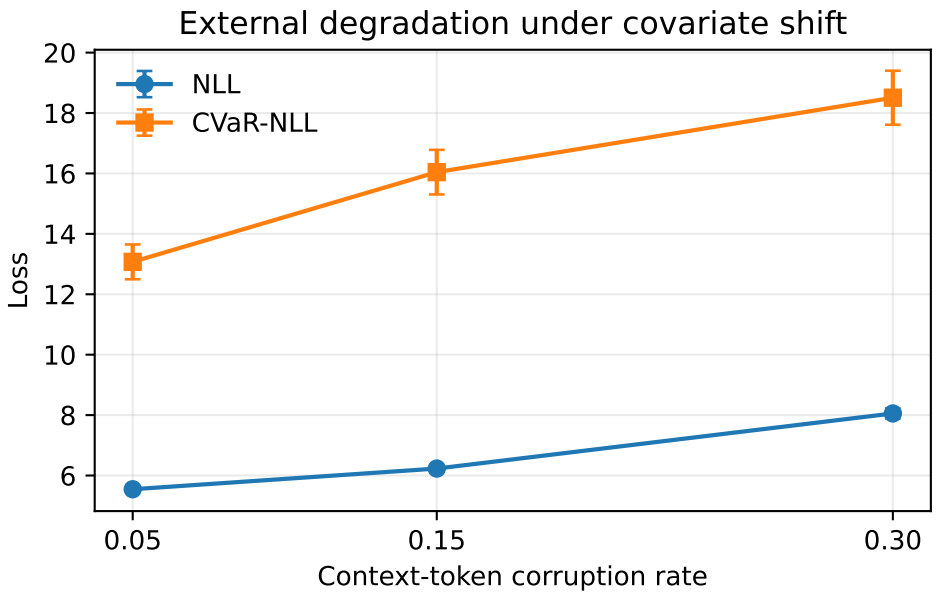

Figure 1: External risk under increasing context-token corruption.

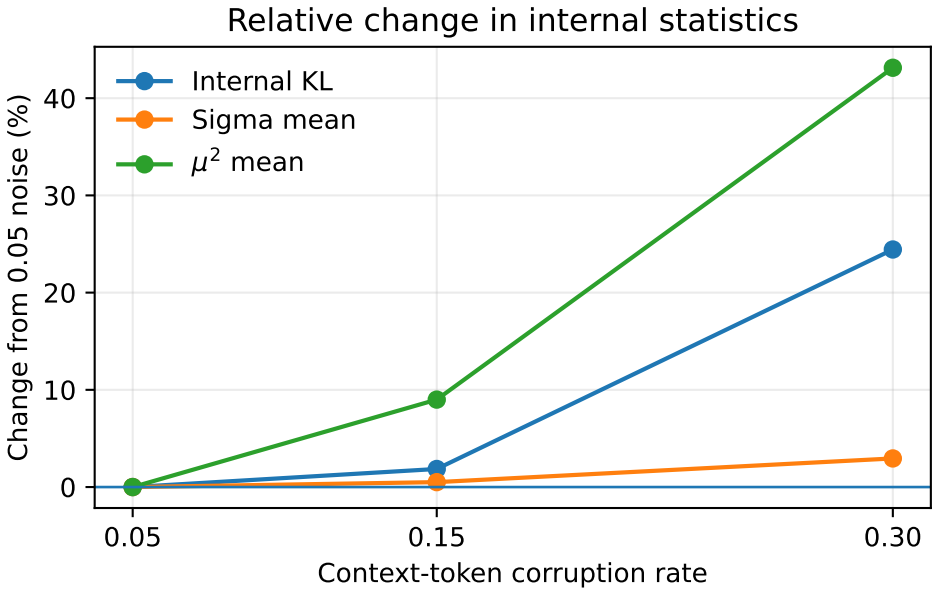

Figure 2: Relative change in local KL, posterior scale, and posterior mean energy under context-token corruption. Mean energy records activation activity, while KL and scale describe the posterior distribution.

Shift family	Internal activity measure	External metric	Pearson r
Covariate	Δ internal KL	Δ NLL	0.852
Covariate	Δ internal KL	Δ CVaR-NLL	0.776
Covariate	Δ posterior mean energy	Δ NLL	0.971
Covariate	Δ posterior mean energy	Δ CVaR-NLL	0.891
Target frequency	Δ internal KL	Δ NLL	0.742
Target frequency	Δ internal KL	Δ CVaR-NLL	0.738
Target frequency	Δ posterior mean energy	Δ NLL	0.929
Target frequency	Δ posterior mean energy	Δ CVaR-NLL	0.928
All shifts	Δ posterior mean energy	Δ CVaR-NLL	0.708

Table 8: Selected internal–external correlations across the evaluated shift conditions. Values summarize aggregate associations between internal activity and external risk.

7 Discussion

7.1 Mechanistic interpretation of the ablations

The ablations separate three roles. Sampling contributes to the learned predictive computation; learned scale shapes the posterior profile and predictive metrics; and KL regularization constrains posterior geometry while interacting with likelihood. Together they provide mechanism-level evidence beyond the complete-architecture comparison.

7.2 Relationship to uncertainty quantification and interpretability

The contribution combines uncertainty quantification with unit-level instrumentation. The posterior belongs to hidden computation, making local distributional quantities observable during prediction. This positioning is summarized as *architectural uncertainty quantification with observable unit-level diagnostics*.

7.3 Monitoring versus intervention

Aggregate shift experiments show that internal activity changes with external risk. This profile supports monitoring, alarms, adaptive sampling, and fallback selection. The internal-score rule in Section 4.7 provides a direct template for organizing these quantities into an operational decision.

7.4 Capacity and stochastic regularization

The model uses rich instrumentation to demonstrate the local posterior mechanism. Parameter and compute accounting places the predictive gains in the context of the complete architecture, and the no-KL, fixed-scale, and mean-only ablations identify distinct roles for its distributional components.

8 Conclusion

Variational neurons provide a concrete mechanism for making probabilistic activity observable inside a language-model computation. In the controlled WritingPrompts experiment, raw EVE improves NLL, perplexity, accuracy, and tail-risk CVaR relative to raw deterministic and Monte Carlo Dropout baselines across five seeds, while exposing local KL, posterior scale, posterior mean energy, and sample disagreement. The ablations show that stochastic sampling is consequential, learned scale changes posterior behavior, and KL regularization controls latent geometry while interacting with predictive likelihood.

Together, the results establish measurable internal probabilistic activity, mechanism-level differentiation among sampling, scale learning, and KL regularization, and aggregate reliability monitoring. The input-dependent score formulation provides an operational template for adaptive sampling, fallback selection, retrieval, and human review.

References

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep Variational Information Bottleneck. In *ICLR*, 2017.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *ICML*, 2015.

Däubener, S., Damm, S., and Fischer, A. ELBO, regularized maximum likelihood, and their common one-sample approximation for training stochastic neural networks. In *UAI*, pp. 897–914, 2025.

Dai, B., Zhu, C., Guo, B., and Wipf, D. Compressing neural networks using the Variational Information Bottleneck. In *ICML*, pp. 1135–1144, 2018.

Damianou, A. and Lawrence, N. D. Deep Gaussian processes. In *AISTATS*, pp. 207–215, 2013.

Elhage, N. et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Eustratiadis, P., Gouk, H., Li, D., and Hospedales, T. Weight-covariance alignment for adversarially robust neural networks. In *ICML*, pp. 3047–3056, 2021.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In *ACL*, 2018.

Frey, B. J. Continuous sigmoidal belief networks trained using slice sampling. In *NeurIPS*, 1996.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. In *NeurIPS*, 2017.

Geifman, Y. and El-Yaniv, R. SelectiveNet: A deep neural network with an integrated reject option. In *ICML*, pp. 2151–2159, 2019.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *EMNLP*, 2021.

Giulianelli, M., Baan, J., Aziz, W., Fernández, R., and Plank, B. What comes next? Evaluating uncertainty in neural text generators against human production variability. In *EMNLP*, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *ICML*, 2017.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. The wake–sleep algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

Hron, J., Matthews, A. G. de G., and Ghahramani, Z. Variational Bayesian dropout: pitfalls and fixes. In *ICML*, pp. 2019–2028, 2018.

Huang, C.-W., Touati, A., Vincent, P., Dziugaite, G. K., Lacoste, A., and Courville, A. Stochastic neural network with Kronecker flow. In *AISTATS*, pp. 4184–4194, 2020.

Huang, X., Li, S., Yu, M., Sesia, M., Hassani, H., Lee, I., Bastani, O., and Dobriban, E. Uncertainty in language models: assessment through rank-calibration. In *EMNLP*, pp. 284–312, 2024.

Kapoor, S., Gruver, N., Roberts, M., Pal, A., Dooley, S., Goldblum, M., and Wilson, A. Calibration-tuning: teaching large language models to know what they don't know. In *UncertaiNLP*, pp. 1–14, 2024.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *ICLR*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational Dropout and the local reparameterization trick. In *NeurIPS*, 2015.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as Gaussian processes. In *ICLR*, 2018.

Liu, X., Fatahi Bayat, F., and Wang, L. Enhancing language model factuality via activation-based confidence calibration and guided decoding. In *EMNLP*, pp. 10436–10448, 2024.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *ICML*, pp. 1708–1716, 2016.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

Malinin, A. and Gales, M. Predictive uncertainty estimation via Prior Networks. In *NeurIPS*, 2018.

McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y., and Bau, D. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In *ICML*, pp. 1791–1799, 2014.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational Dropout sparsifies deep neural networks. In *ICML*, pp. 2498–2507, 2017.

Nalisnick, E., Hernández-Lobato, J. M., and Smyth, P. Dropout as a structured shrinkage prior. In *ICML*, pp. 4712–4722, 2019.

Ovadia, Y. et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. OpenAI technical report, 2019.

Ravanbakhsh, S., Poczos, B., Schneider, J., Schuurmans, D., and Greiner, R. Stochastic neural networks with monotonic activation functions. In *AISTATS*, pp. 809–818, 2016.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pp. 1278–1286, 2014.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?": Explaining the predictions of any classifier. In *KDD*, 2016.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Shekhovtsov, A., Yanush, V., and Flach, B. Path sample-analytic gradient estimators for stochastic binary networks. In *NeurIPS*, 2020.

Sun, S., Chen, C., and Carin, L. Learning structured weight uncertainty in Bayesian neural networks. In *AISTATS*, pp. 1283–1292, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *ICML*, 2017.

Tang, C. and Salakhutdinov, R. Learning stochastic feedforward neural networks. In *NeurIPS*, 2013.

Urban, S., Basalla, M., and van der Smagt, P. Gaussian Process Neurons learn stochastic activation functions. arXiv:1711.11059, 2017.

Zhai, Z. Cell Variational Information Bottleneck Network. In *ACML*, pp. 1606–1621, 2024.