Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

The Neuron Is the Distribution

Yves Ruffenach

yves@ruffenach.net

Conservatoire National des Arts et Métiers https://orcid.org/0009-0009-4737-0555

Research Article

Keywords: variational neurons, probabilistic neural networks, variational inference, uncertainty estimation, stochastic hidden representations, deep learning, Transformer language modeling

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10333402/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

The Neuron Is the Distribution

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

Abstract

Neural networks mainly compute with point-valued hidden activations, while uncertainty is usually added around this computation through weights, outputs, ensembles, perturbations, or global latents. We study the neuron itself as a local variational hidden unit.

We introduce Elemental Variational Expanse (EVE), a variational hidden unit with input-dependent posterior inference, reparameterized sampling, neuron-level Kullback–Leibler (KL) regularization, and posterior-derived diagnostics. Unlike a deterministic mean-variance head, which describes a predictive distribution after hidden computation, EVE computes through sampled latent states inside hidden units.

We evaluate EVE through staged evidence: a controlled heteroscedastic regression study for attribution, tabular regression studies for multilayer perceptron (MLP) feasibility, and Transformer language-modeling experiments for feed-forward-block feasibility. Across matched protocols, the largest gains appear on uncertainty-sensitive metrics, while point metrics remain close or improve in several aggregates. The results support a focused concept-and-feasibility claim: EVE trains end-to-end inside ordinary neural architectures, improves probabilistic prediction in controlled settings, and exposes measurable hidden-state diagnostics.

1 Introduction

Neural networks usually compute with point-valued hidden activations: a hidden unit receives an input representation h and emits a deterministic activation. We study an alternative in which the hidden unit itself performs local variational computation.

We introduce Elemental Variational Expanse (EVE), a variational hidden unit with input-dependent posterior inference, reparameterized sampling, neuron-level KL regularization, and diagnostics. A deterministic hidden unit computes

$$a = f(h),$$

where h is the input representation and a is the emitted activation. An EVE unit instead constructs

$$q_\phi(z \mid h) = \mathcal{N}\big(\mu_\phi(h), \operatorname{diag}(\sigma_\phi^2(h))\big),$$

samples

$$z = \mu_\phi(h) + \sigma_\phi(h) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

and produces the activation from z. Thus, hidden computation propagates a sampled latent state rather than only a deterministic activation.

Terminology. We use the following terms in a narrow technical sense. A *variational neuron* is a hidden unit that replaces a deterministic activation with an input-dependent latent posterior, reparameterized sampling, and a local KL penalty. By *neuron-level variational computation*, we mean that sampling and KL regularization occur inside hidden units rather than only at the output or parameter level. By *diagnostics*, we mean posterior-derived quantities measured during

training or evaluation, including local KL, posterior variance, posterior scale, posterior-mean energy $\frac{1}{d_z}\sum_j \mu_j^2$, and target-band occupancy over a predefined interval $[l, u]$.

This differs from output-level uncertainty. A deterministic mean-variance head describes a predictive distribution after hidden computation, whereas EVE samples and regularizes latent states inside hidden units. In this sense, a mean-variance head describes a distribution, while EVE computes through one.

We evaluate this claim through three stages: controlled attribution in a heteroscedastic regression study, tabular MLP feasibility on BostonHousing and ConcreteStrength, and Transformer feed-forward feasibility on WritingPrompts and WikiText2 raw.

Contributions. We make three contributions. First, we introduce EVE, a reusable variational hidden-unit module with posterior inference, local prior regularization, reparameterized sampling, neuron-level KL, and diagnostics. Second, we distinguish output-level uncertainty from stochastic computation inside hidden units. Third, we provide staged empirical evidence: a controlled heteroscedastic study separating parameter count, output variance, and neuron-level variational computation; tabular regression studies testing MLP feasibility; and Transformer language-modeling experiments testing feed-forward-block feasibility.

2 Related Work

Parameter-level and latent-variable uncertainty. Bayesian neural networks place uncertainty over parameters [5,1]. Variational inference gives scalable approximations for probabilistic deep learning [1]. Latent-variable models use variational posteriors and reparameterized sampling to learn stochastic representations [3,6]. These methods usually place uncertainty at the parameter, data-representation, global-structure, or encoder–decoder level. EVE uses a different granularity: it moves posterior inference, sampling, KL regularization, and diagnostics to the hidden unit itself.

Output uncertainty, ensembles, and perturbations. Many predictors express output uncertainty through means, variances, quantiles, likelihood losses, or calibration methods [2]. Heteroscedastic regressors predict input-dependent output variance as a form of output-level uncertainty. Deep ensembles, dropout, and stochastic perturbations improve predictive uncertainty by combining predictions or injecting randomness [4,7]. These methods usually describe uncertainty around or after deterministic hidden computation. EVE instead places stochastic inference and regularization inside the hidden unit.

Stochastic hidden representations and Transformer computation. Several methods move stochasticity toward hidden representations. Variational dropout and the local reparameterization trick translate weight uncertainty into local activation noise independent across minibatch examples [11]. Dropout-based Bayesian approximations and concrete dropout use stochastic masks or learned dropout rates for uncertainty and regularization [7,12]. Information Dropout and the Deep Variational Information Bottleneck use noisy or variational representations to control information flow [13,14]. Stochastic neural networks model activation uncertainty directly and can improve robustness, pruning, or uncertainty behavior [15].

EVE is closest to stochastic hidden-representation methods, but differs in the object being modeled. Variational dropout expresses weight-induced or multiplicative noise. Information-bottleneck methods regularize representation capacity. Stochastic hidden-state methods inject or learn activation noise. EVE instead makes each hidden unit a local variational object: it has an input-dependent posterior, standard prior, reparameterized sample, neuron-level KL, and posterior-derived diagnostics. Thus, the unit is not only noisy or compressed; it is the distributional computational object. We test this primitive in dense MLPs and Transformer

feed-forward blocks [8], with WritingPrompts as the main Transformer feasibility setting [10] and WikiText2 raw as a standard early-epoch validation corpus [9].

3 Method

3.1 Variational neuron

Let h be the input representation of a hidden unit. A deterministic unit computes

$$a = f_\theta(h).$$

EVE replaces this point computation with

$$q_\phi(z \mid h) = \mathcal{N}\big(\mu_\phi(h), \mathrm{diag}(\sigma_\phi^2(h))\big), \qquad p(z) = \mathcal{N}(0, I),$$
$$z = \mu_\phi(h) + \sigma_\phi(h) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

and emits

$$a = g_\theta(z, h).$$

This gives each unit a posterior, prior, sampled state, local KL term, and measurable diagnostics.

3.2 Objective and diagnostics

Training combines task loss and neuron-level KL,

$$\mathrm{KL}\left(q_\phi(z \mid h) \,\|\, p(z)\right).$$

For a diagonal Gaussian posterior and standard Gaussian prior,

$$\mathrm{KL}(q\|p) = \frac{1}{2} \sum_j \left(\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1\right).$$

The objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \beta \mathcal{L}_{\text{KL}}.$$

Each EVE unit exposes posterior-mean energy,

$$\mu^2 = \frac{1}{d_z} \sum_{j=1}^{d_z} \mu_j^2,$$

local KL, posterior scale, and target-band occupancy statistics. For target band $[l, u]$, occupancy reports below-band, inside-band, and above-band fractions. These diagnostics summarize the internal posterior state and can be monitored during training, used for model selection, or used by future regulation mechanisms.

3.3 Baselines and evaluation

We compare EVE with baselines matched whenever possible in data split, architectural placement, optimizer family, training budget, predictive head, evaluation pipeline, and metrics. The controlled heteroscedastic study separates three controls: a scalar deterministic baseline (DET-scalar), a deterministic capacity-matched cell (DET-cell), and a stochastic heteroscedastic output baseline (STOCH-hetero). Regression uses mean squared error (MSE), mean absolute error (MAE), negative log-likelihood (NLL), Monte Carlo negative log-likelihood (MC-NLL), continuous ranked probability score (CRPS), and pinball loss when available. Language modeling uses cross-entropy (CE), perplexity (PPL), accuracy, MC-NLL, expected calibration error (ECE), conditional-value-at-risk negative log-likelihood (CVaR-NLL), mutual information (MI), and top-1 flip rate. Lower is better except for accuracy; MI and flip rate are reported only as sampling-based uncertainty diagnostics.

4 Experiments

4.1 Protocol overview

The experiments follow three stages. First, a controlled heteroscedastic scalar regression study provides the attribution test, using 5 seeds and comparing DET, DET-cell, STOCH-hetero, and EVE with validation selection and held-out test reporting. Second, tabular MLP feasibility is tested on BostonHousing with 10 seeds and on ConcreteStrength with 10 seeds and a 720/103/207 train/validation/test split; both use validation-selected models and held-out test evaluation. Third, Transformer language-modeling feasibility is tested with paired validation protocols: WritingPrompts uses 3 seeds and approximately 3.08×10^5 training targets, while WikiText2 raw uses 3 seeds, 1.03×10^6 training windows, and an early-epoch validation protocol.

4.2 Controlled heteroscedastic regression study

We start with a controlled heteroscedastic regression study designed to test whether observed gains come from parameter count, deterministic internal capacity, output-level heteroscedastic variance, or neuron-level variational computation. Because the task has a known heteroscedastic regime, both predictive uncertainty and internal diagnostics can be compared with the true conditional standard deviation.

The comparison includes DET-scalar, DET-cell, STOCH-hetero, and EVE. DET-cell controls parameter count and deterministic internal capacity; STOCH-hetero controls output-level uncertainty. EVE is therefore evaluated against both a stronger deterministic cell and a direct output-variance alternative.

We use the label

$$\text{EVE-}k2_\text{soft-band}_\text{dual-point}_\text{point025}$$

only as a configuration identifier. Here, $k = 2$ is the latent dimension, *soft-band* denotes a penalty encouraging posterior-mean energy to remain in a target interval, *dual-point* denotes the auxiliary point-prediction path, and *point025* denotes a point-loss weight of 0.25. The model uses hidden width 40, $\beta = 10^{-3}$, and band weight 0.05. Model selection is point-first probabilistic validation selection with guard tolerance 0.020. The deterministic capacity-matched cell uses hidden width 41 and 4109 trainable parameters. The selected EVE model uses 3970 trainable parameters. Thus the deterministic matched cell has more parameters than EVE.

Table 1: Controlled heteroscedastic study test metrics. Values are mean ± standard deviation across paired seeds. Lower is better. EVE approaches the deterministic capacity-matched MSE within 0.5%, while substantially improving NLL, MC-NLL, and CRPS.

Model	MSE ↓	NLL ↓	MC-NLL ↓	CRPS ↓
DET-cell h41	$\mathbf{0.057833 \pm 0.007422}$	0.000393 ± 0.069350	-0.005816 ± 0.067101	0.121007 ± 0.004508
DET-scalar	0.059886 ± 0.006961	0.020700 ± 0.062816	0.012061 ± 0.059506	0.124654 ± 0.003804
EVE dual point025	0.058069 ± 0.007376	$\mathbf{-0.402780 \pm 0.037055}$	$\mathbf{-0.431748 \pm 0.033447}$	$\mathbf{0.109431 \pm 0.004106}$
STOCH-hetero	0.061371 ± 0.006984	-0.383516 ± 0.032522	-0.390487 ± 0.033141	0.112468 ± 0.003594

EVE reaches an MSE of 0.058069, compared with 0.057833 for the deterministic capacity-matched cell, a relative gap of approximately 0.41%:

$$0.058069 - 0.057833 = 0.000236.$$

It improves over that cell on NLL, MC-NLL, and CRPS. Against STOCH-hetero, EVE improves all aggregate metrics: MSE 0.061371 → 0.058069, NLL −0.383516 → −0.402780, MC-NLL −0.390487 → −0.431748, and CRPS 0.112468 → 0.109431. Relative to STOCH-hetero, this gives a 5.38% MSE reduction and a 2.70% CRPS reduction; EVE also improves over DET-scalar on every metric. Together, these comparisons support the attribution claim: the gain is not explained only by deterministic capacity or output-level heteroscedastic variance.

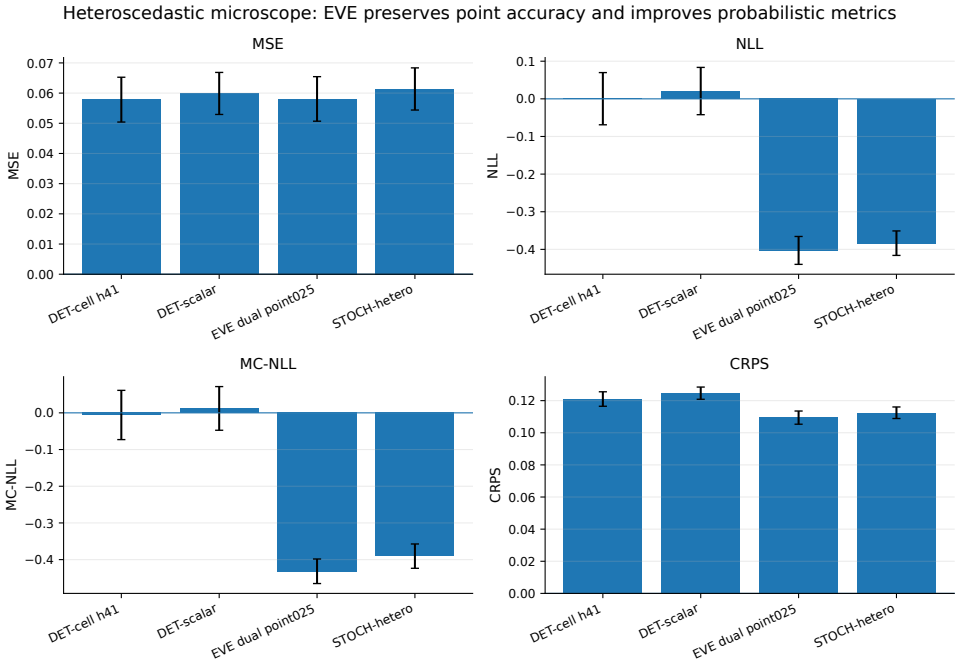


Figure 1: Controlled heteroscedastic study main metrics. EVE stays within a small point-accuracy gap relative to the deterministic capacity-matched cell while improving uncertainty-sensitive metrics. The hatched bars mark the best value in each panel.

Diagnostics and cost. The known heteroscedastic regime also lets us evaluate whether internal diagnostics track the true conditional standard deviation.

Table 2: Tracking of the true heteroscedastic regime. EVE tracks the true conditional standard deviation almost as strongly as the heteroscedastic output baseline, while outperforming it on all aggregate predictive metrics.

Model	Corr. pred std / true std ↑	Pred std mean
DET-cell h41	0.016264 ± 0.014075	0.230284 ± 0.009190
DET-scalar	0.017618 ± 0.037545	0.237716 ± 0.007205
EVE dual point025	0.980922 ± 0.005937	0.221741 ± 0.011905
STOCH-hetero	$\mathbf{0.990188 \pm 0.005011}$	0.200247 ± 0.004043

STOCH-hetero has the highest direct correlation between predicted and true standard deviation, and EVE is close. The deterministic references remain close to constant predictive-scale behavior. EVE therefore tracks the regime while also improving over STOCH-hetero on MSE, NLL, MC-NLL, and CRPS.

EVE also exposes posterior-derived signals aligned with the true regime. Posterior variance is the strongest diagnostic. Local KL and posterior-mean energy also correlate with the true conditional standard deviation.

These results show that uncertainty-relevant structure is visible inside the hidden computation, not only in the predictive output scale.

EVE costs 4.04× the scalar deterministic baseline in MC forward time and 1.77× the deterministic capacity-matched cell. This cost comes from posterior parameterization, latent sampling, and probabilistic evaluation. The capacity comparison remains conservative because DET-cell h41 has more parameters than EVE.

Table 3: EVE diagnostics in the controlled heteroscedastic study. Posterior variance, local KL, and posterior-mean energy correlate with the known true conditional standard deviation. These diagnostics are internal to the hidden computation.

EVE diagnostic	Corr. with true std ↑	Mean value
Posterior variance σ_z^2	$\mathbf{0.923168 \pm 0.078822}$	1.026079 ± 0.154231
Local KL	0.847664 ± 0.145940	0.657025 ± 0.217884
Posterior-mean energy μ^2	0.770044 ± 0.226064	0.609853 ± 0.273000

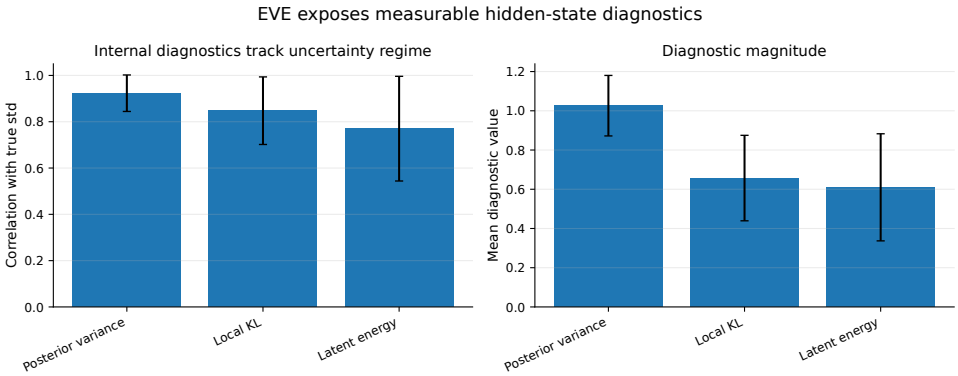


Figure 2: EVE diagnostics in the controlled heteroscedastic study. Posterior variance, local KL, and posterior-mean energy correlate with the known conditional standard deviation, showing that uncertainty-relevant structure is visible inside the hidden computation.

Table 4: Controlled heteroscedastic study computational profile. The deterministic capacity-matched cell has more trainable parameters than EVE, giving a conservative capacity comparison. EVE has higher sampling cost, as expected for local variational computation.

Model	Params	State MB	Forward MC ms	Train step ms	Forward ratio
DET-scalar	5	0.000019	0.238570	1.953486	1.00×
STOCH-hetero	1,766	0.006737	0.389760	2.452732	1.63×
DET-cell h41	4,109	0.015675	0.543710	2.983544	2.28×
EVE dual point025	3,970	0.015144	0.963730	5.616957	4.04×

4.3 BostonHousing feasibility

BostonHousing tests whether EVE can be inserted into a small real-data tabular regression pipeline. We compare EVE with a deterministic probabilistic baseline (DET-Prob) under the same train/validation/test protocol, backbone size, training budget, and metrics. Only the hidden unit changes.

The selected EVE configuration uses latent dimension $k = 4$, an adaptive KL coefficient, an input-dependent output scale, Monte Carlo likelihood evaluation, local posterior and decoder maps, and post-hoc output-scale calibration. Calibration rescales predictive standard deviation using validation data while leaving the predictive mean unchanged. The calibrated row isolates output-scale calibration: it changes predictive variance but leaves the predictive mean, MSE, and MAE unchanged. The main architectural evidence remains the controlled heteroscedastic study and the repeated-seed ConcreteStrength comparison.

Table 5: BostonHousing test performance over ten paired seeds. Both EVE rows use the same validation-selected checkpoint. Calibration changes only sigma, so MSE and MAE are identical. Values are mean ± standard deviation.

Model	MSE ↓	MAE ↓	NLL ↓	CRPS ↓	Pinball ↓
EVE raw	$\mathbf{0.1374 \pm 0.0524}$	$\mathbf{0.2431 \pm 0.0298}$	1.2032 ± 0.7136	0.1847 ± 0.0251	0.0849 ± 0.0121
EVE calibrated	$\mathbf{0.1374 \pm 0.0524}$	$\mathbf{0.2431 \pm 0.0298}$	$\mathbf{0.3055 \pm 0.1737}$	$\mathbf{0.1793 \pm 0.0233}$	$\mathbf{0.0816 \pm 0.0114}$
DET-Prob	0.1845 ± 0.0813	0.2603 ± 0.0345	0.4812 ± 0.2634	0.1908 ± 0.0245	0.0857 ± 0.0108

Calibrated EVE improves over DET-Prob on all aggregate metrics: MSE 0.1845 → 0.1374, MAE 0.2603 → 0.2431, NLL 0.4812 → 0.3055, CRPS 0.1908 → 0.1793, and pinball 0.0857 → 0.0816. Relative reductions are 25.53%, 6.61%, 36.51%, 6.03%, and 4.78%. Calibration preserves MSE and MAE, and improves NLL 1.2032 → 0.3055, CRPS 0.1847 → 0.1793, and pinball 0.0849 → 0.0816.

Calibrated EVE wins against DET-Prob on MSE 8/10, MAE 8/10, NLL 7/10, CRPS 7/10, pinball 6/10, and MSE–NLL–CRPS jointly 6/10. This stage shows that EVE can be inserted into a real tabular regression network, trained end-to-end, selected on validation only, calibrated without changing the predictive mean, and used for prediction.

4.4 ConcreteStrength dense MLP benchmark

ConcreteStrength is the main repeated-seed dense MLP benchmark, following the controlled attribution study and BostonHousing feasibility test. It has 8 inputs, one scalar target, and 720/103/207 train/validation/test examples. DET and EVE share the same dense MLP family, split, optimizer family, training budget, predictive head, evaluation pipeline, and metrics. Only the hidden unit changes: DET uses point activations, while EVE uses local posterior inference, reparameterized sampling, neuron-level KL, and diagnostics.

Table 6 reports the main dense MLP result. Across 10 paired seeds, validation-selected EVE improves all aggregate test metrics over DET. MSE decreases 0.9760 → 0.9715, NLL 3.8328 → 3.1485, and CRPS 0.6449 → 0.6287. Relative reductions are 0.46%, 17.85%, and 2.51%. EVE wins NLL and CRPS in all 10 paired seeds, MSE in 7/10, and all three metrics jointly in 7/10.

Table 6: ConcreteStrength main repeated-seed result. DET and the selected EVE prediction rule share the same dense MLP protocol. Lower is better for all metrics.

Model	MSE ↓	NLL ↓	CRPS ↓
DET	0.9760	3.8328	0.6449
EVE selected predictor	**0.9715**	**3.1485**	**0.6287**
Relative reduction	0.46%	17.85%	2.51%
EVE seedwise wins	7/10	10/10	10/10

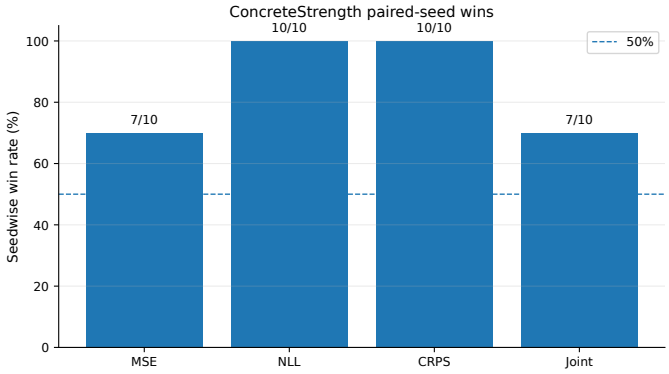


Figure 3: ConcreteStrength paired-seed win rates. EVE wins NLL and CRPS in all ten paired seeds, MSE in seven out of ten, and all three metrics jointly in seven out of ten.

4.5 Transformer language modeling

We test EVE inside Transformer-style next-token prediction. WritingPrompts is the main paired Transformer comparison. WikiText2 raw is a standard early-epoch validation check.

WritingPrompts. The WritingPrompts experiment uses WritingPrompts-Filtered with GPT-style tokenization and teacher-forcing windows. For each paired seed, DET and EVE use the same shuffled split, tokenizer, batch size, training horizon, language-modeling head, and validation pipeline. Only the feed-forward computational primitive changes. Each run uses 3,188 training sequences, 797 validation sequences, about 3.08×10^5 training targets, and 7.7×10^4 validation targets.

Tables 7 and 8 report three-seed validation aggregates. EVE improves CE 5.1604 → 4.9916, PPL 174.23 → 147.28, accuracy 0.2022 → 0.2174, and MC-NLL 5.2570 → 5.1016. Because EVE samples hidden states, repeated forward passes also produce nonzero sampling-based uncertainty diagnostics.

Table 7: WritingPrompts Transformer validation task metrics over three paired seeds. Values are mean ± standard deviation.

Model	CE ↓	PPL ↓	Acc ↑
DET	5.1604 ± 0.0091	174.23 ± 1.58	0.2022 ± 0.0010
EVE	$\mathbf{4.9916 \pm 0.0467}$	$\mathbf{147.28 \pm 6.96}$	$\mathbf{0.2174 \pm 0.0018}$

Seed-wise WritingPrompts validation is consistent. EVE wins CE, perplexity, and accuracy on 3/3 paired seeds, with CE 5.1598 → 4.9565, 5.1697 → 5.0446, and 5.1516 → 4.9737 for seeds 0–2. The mean CE gain is 0.1688, the mean perplexity gain is 26.95, and the mean accuracy gain is 0.0152. ECE is lower for EVE on average, while CVaR-NLL favors DET on the mean. Under repeated deterministic evaluation, DET has zero top-1 flip rate and near-zero mutual information. EVE produces mean mutual information 0.1592 and mean top-1 flip rate 0.2958.

Table 8: WritingPrompts Transformer validation uncertainty metrics over three paired seeds. Values are mean ± standard deviation. Lower is better for MC-NLL, ECE, and CVaR-NLL. MI and flip rate are reported as sampling-based uncertainty signals.

Model	MC-NLL ↓	ECE ↓	MI	Flip	CVaR-NLL ↓
DET	5.2570 ± 0.0806	0.0431 ± 0.0021	≈ 0	0.0000	$\mathbf{12.6675 \pm 0.2372}$
EVE	$\mathbf{5.1016 \pm 0.1367}$	$\mathbf{0.0406 \pm 0.0116}$	$\mathbf{0.1592 \pm 0.0361}$	$\mathbf{0.2958 \pm 0.0406}$	12.9347 ± 0.4311

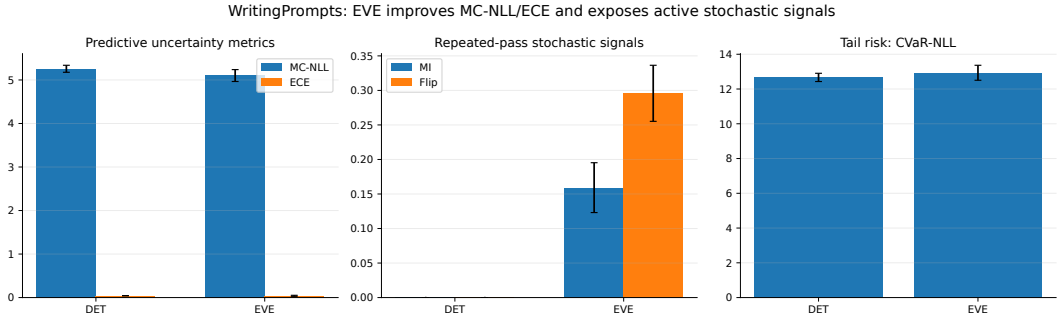


Figure 4: WritingPrompts uncertainty and stochasticity diagnostics. EVE improves MC-NLL and ECE on average and exposes repeated-pass stochastic signals, while deterministic repeated evaluation has near-zero mutual information and zero top-1 flip rate.

WritingPrompts internal diagnostics show that the stochastic hidden state remains active rather than collapsing: latent KL 0.4180 ± 0.0620, posterior-mean energy $\mu^2 = 0.1483 \pm 0.0171$, active units 1.0000 ± 0.0000, active dimensions 1024.0 ± 0.0, adaptive $\beta = 2.56 \times 10^{-4} \pm 9.06 \times 10^{-5}$, μ^2 exponential moving average (EMA) 0.1343 ± 0.0068, KL EMA 0.2391 ± 0.0094, and alive EMA 1.0000 ± 0.0000.

WikiText2 raw. WikiText2 raw complements WritingPrompts with a standard language-modeling corpus and a short early-epoch validation protocol. The same paired-control rule is used: the split, tokenization, training horizon, validation procedure, and next-token objective are fixed, and only the feed-forward computational primitive changes. The comparison uses three paired seeds and four epochs, with 1,027,552 supervised training windows and 108,027 validation windows.

Table 9: WikiText2 raw early-epoch Transformer validation comparison over three paired seeds. Only the first four epochs are considered. Values are mean ± standard deviation across seeds.

Epoch	Model	CE ↓	PPL ↓	Acc ↑
1	DET	5.3449 ± 0.0521	209.73 ± 11.06	0.2196 ± 0.0066
1	EVE	$\mathbf{5.2797 \pm 0.0982}$	$\mathbf{196.93 \pm 18.82}$	$\mathbf{0.2257 \pm 0.0040}$
2	DET	5.2070 ± 0.0822	182.95 ± 15.35	0.2257 ± 0.0109
2	EVE	$\mathbf{5.1396 \pm 0.0748}$	$\mathbf{170.96 \pm 12.53}$	$\mathbf{0.2422 \pm 0.0162}$
3	DET	5.2084 ± 0.0560	183.00 ± 10.40	0.2301 ± 0.0091
3	EVE	$\mathbf{5.1060 \pm 0.0225}$	$\mathbf{165.04 \pm 3.73}$	$\mathbf{0.2387 \pm 0.0015}$
4	DET	5.3752 ± 0.0355	216.08 ± 7.61	0.2188 ± 0.0120
4	EVE	$\mathbf{5.0420 \pm 0.0537}$	$\mathbf{154.92 \pm 8.19}$	$\mathbf{0.2517 \pm 0.0054}$

Across four epochs, EVE improves validation CE, perplexity, and accuracy relative to DET. At epoch 4, EVE reaches 5.0420 ± 0.0537 CE, 154.92 ± 8.19 PPL, and 0.2517 ± 0.0054 accuracy, compared with 5.3752 ± 0.0355, 216.08 ± 7.61, and 0.2188 ± 0.0120 for DET. The epoch-4 gains are 0.3332 CE, 61.16 PPL, and 0.0329 accuracy; EVE wins all three metrics on all three paired seeds.

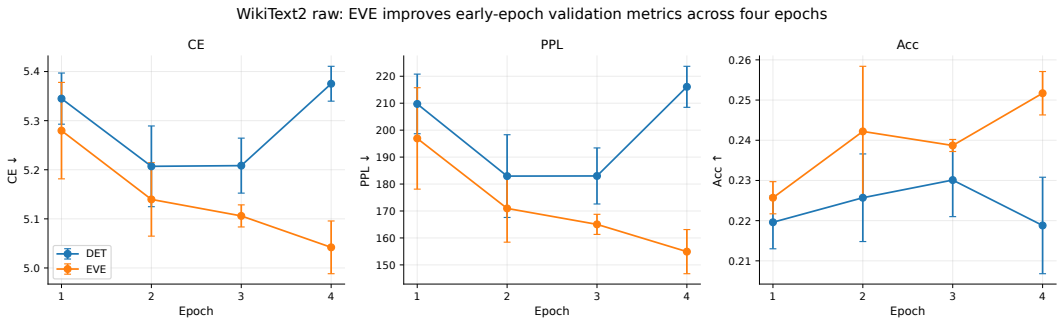


Figure 5: WikiText2 raw early-epoch validation curves. Across the first four epochs, EVE improves validation CE, perplexity, and accuracy relative to the matched deterministic Transformer feed-forward baseline.

Table 10: WikiText2 raw epoch-4 train–validation gap comparison. Values are means over three seeds. EVE has the lower validation CE and the smaller validation gap at epoch 4.

Model	Train CE epoch 4	Val CE epoch 4	Gap
DET	4.3317	5.3752	1.0435
EVE	4.9394	**5.0420**	**0.1026**

At epoch 4, DET has lower training CE and a larger train–validation gap, 1.0435. EVE has higher training CE and lower validation CE, with gap 0.1026. In this short validation regime, the larger training CE but lower validation CE suggests that variational feed-forward computation acts as an internal regularizer. EVE also exposes stable internal diagnostics: the aggregate diagnostic KL decreases from 428.30 ± 20.52 to 260.43 ± 10.65 over epochs 1–4, while the normalized KL regularization term stabilizes around 0.168, and target-band activity remains active.

5 Discussion

The staged results support EVE as a neuron-level variational computational primitive rather than a post-hoc uncertainty head. The controlled heteroscedastic study provides the attribution test: EVE approaches the deterministic capacity-matched MSE within 0.5%, despite using fewer parameters, improves over the heteroscedastic output control on MSE, NLL, MC-NLL, and CRPS, and exposes diagnostics aligned with the known heteroscedastic regime. The behavior is therefore visible inside hidden computation, not only at the predictive output. The later stages test feasibility across architectures. BostonHousing shows small real-data tabular feasibility. ConcreteStrength provides the main repeated-seed dense MLP result, with NLL and CRPS wins in every paired seed and MSE wins in seven out of ten. The Transformer experiments show that the same primitive remains trainable in feed-forward blocks, improving validation metrics on WritingPrompts and early-epoch WikiText2 raw. The claim is focused: EVE is not presented as a universal accuracy improvement. It is evidence that hidden units can perform local variational computation, expose measurable posterior-state diagnostics, and improve uncertainty-sensitive prediction under matched protocols.

6 Scope and Future Work

This work is bounded concept-and-feasibility evidence, not a claim of universal accuracy improvement or compute-normalized dominance. EVE adds cost through posterior parameterization, latent sampling, local KL, and probabilistic evaluation; in the controlled heteroscedastic study, it

costs 4.04× the scalar deterministic baseline in MC forward time and 1.77× the deterministic capacity-matched cell. Future work should target lower sampling cost, broader architectures, larger datasets, compute-normalized protocols, additional matched-capacity controls, calibration-matched baselines, and systematic controllers.

7 Conclusion

We introduced Elemental Variational Expanse (EVE), a local variational neuron with posterior inference, reparameterized sampling, neuron-level KL regularization, and diagnostics. Across staged experiments, EVE separates neuron-level variational computation from parameter count and output variance, preserves close or favorable point accuracy, improves uncertainty-sensitive metrics, and exposes measurable hidden-state diagnostics. These results show that neuron-level variational computation can train end-to-end inside ordinary neural architectures and make uncertainty part of hidden computation itself.

Competing Interests

The author declares no competing interests.

References

[1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1613–1622, 2015.

[2] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330, 2017.

[3] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[4] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

[5] Radford M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer, 1996.

[6] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059, 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[9] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

[10] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, 2018.

[11] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, 2015.

[12] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, 2017.

[13] Alessandro Achille and Stefano Soatto. Information dropout: learning optimal representations through noisy computation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(12):2897–2905, 2018.

[14] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[15] Tianyuan Yu, Yongxin Yang, Da Li, Timothy Hospedales, and Tao Xiang. Simple and effective stochastic neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[16] David Harrison Jr. and Daniel L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. doi:10.1016/0095-0696(78)90006-2.

[17] Scikit-learn developers. `sklearn.datasets.load_boston`: dataset documentation and ethical warning. Scikit-learn documentation, version 1.1.3, 2022.

[18] I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. doi:10.24432/C5PK67.

[19] I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998. doi:10.1016/S0008-8846(98)00165-3.

A Dataset Provenance and Availability

This study uses existing public benchmark datasets only. No new dataset is introduced, no raw dataset is redistributed, and no supplementary files accompany this preprint. The datasets were obtained from their original public sources or from standard public dataset interfaces, subject to the applicable terms of their providers.

Synthetic heteroscedastic microscope. The controlled heteroscedastic regression data are generated synthetically for this study and contain no third-party data.

BostonHousing. BostonHousing is used only as a small historical benchmark for initial feasibility testing. The dataset originates from the Boston metropolitan-area housing data studied by Harrison and Rubinfeld [16]. It contains an ethically problematic engineered variable based on racial composition and has been deprecated for general machine-learning use [17]. Accordingly, it is used here only for numerical benchmarking and not for social inference, deployment, or decision-making.

ConcreteStrength. ConcreteStrength is used as the main repeated-seed dense MLP regression benchmark. The UCI Concrete Compressive Strength dataset contains 1,030 examples, eight quantitative input variables, and one quantitative target [18]. It was created by I-Cheng Yeh and is associated with the original study on neural-network modeling of concrete strength [19]. The UCI Machine Learning Repository distributes the dataset under the Creative Commons Attribution 4.0 International license and assigns it the DOI `https://doi.org/10.24432/C5PK67`.

WritingPrompts. WritingPrompts is used for the main paired Transformer feasibility experiment. The dataset consists of writing prompts paired with human-written stories from an online forum [10]. The raw corpus is not redistributed with this preprint and remains subject to the terms of its original source.

WikiText2 raw. WikiText2 raw is used as a corroborating language-modeling benchmark. The corpus is derived from Wikipedia articles and was introduced by Merity et al. [9]. The raw dataset is not redistributed with this preprint and remains subject to the terms of its original release and source material.