Bayesian Backprop as Belief Propagation

Single-Pass Predictive Uncertainty

Wen Dong, Air Force Research Laboratory NeurIPS 2026

Current uncertainty practices

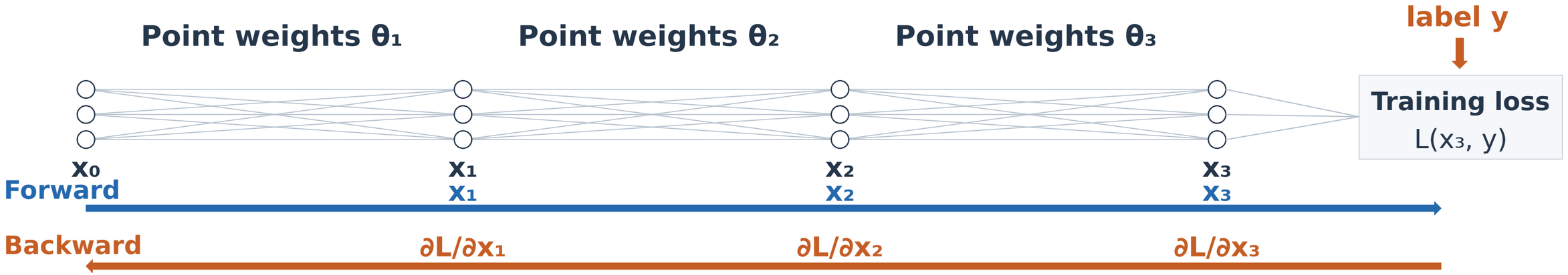
Practice	Learning and prediction
Temperature scaling	Recalibrate a trained model’s confidence scores.
Deep ensembles	Train several networks and combine their predictions.
MC dropout	Repeat prediction with different dropout masks.
VI, Laplace, SWAG	Approximate weight beliefs, often sampling for prediction.
Single-pass methods	Uncertainty-aware heads and existing moment propagation.

Repeated model evaluations can make uncertainty costly to serve.

Bayesify converts selected neural modules into moment-propagation modules and generates their belief-update rules.

Ordinary forward–backward propagation

Activations move forward. Loss derivatives return to the weights.



The backward pass also supplies weight derivatives ∇θL.

Example update (α: step size) $$\theta^{+} = \theta - \alpha \nabla_{\theta} L$$

Operation	Ordinary network
Learned state	Point-valued weights θ
Forward pass	Activations x
Backward pass	Loss derivatives used by the optimizer

Prediction uses only the forward pass.

Backpropagation of posterior corrections

Theorem 3.1: exact normalized junction trees, with represented messages and retained normalizers

$$d_p = F_p g_p^{\mathrm{loc}} + \sum_c F_p J_{c,p}^{\top} F_c^{-1} d_c$$

$$\mu_p^{\mathrm{post}} = \mu_p + d_p$$

posterior = reference + correction

$g = \nabla_\mu \log Z, \qquad d = Fg$

p: parent c: child

$J_{c,p} = \dfrac{\partial \mu_c}{\partial \mu_p}$

Z: evidence normalizer

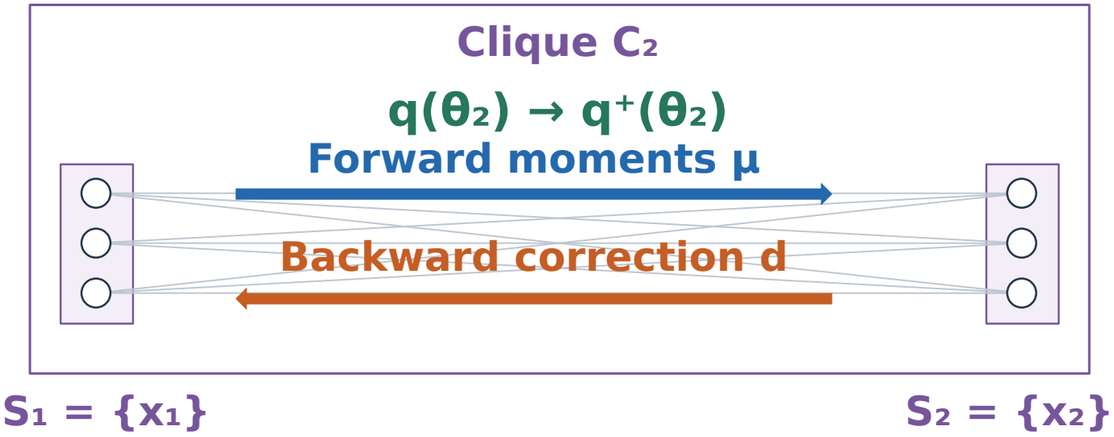
d: moment pseudo-gradient

forward moment Jacobian

Local term: direct evidence/normalizer derivatives. F: covariance of retained statistics (Fisher matrix).

Proof: differentiate the tree-cut evidence, then backpropagate through exact products, marginals and normalizers.

A local part of the neural moment graph



p: input or weight belief c: output belief

	Ordinary	Bayesified
State	Point θ	Belief q(θ)
Forward	Values x	Moments μ
Backward	Derivatives	Corrections d

The neural graph uses projected moments.
The next slide gives its EP update.

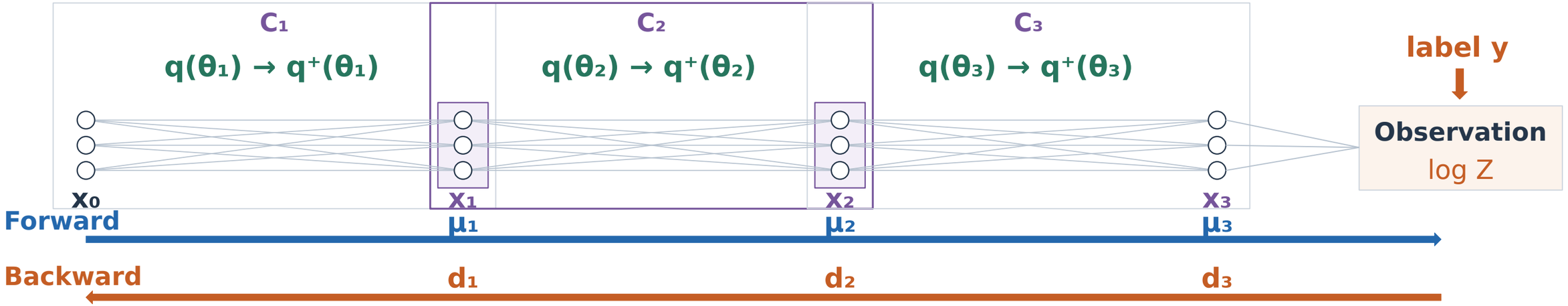
The EP Bayesifier

Theorem 3.2: local tilted moments from reverse derivatives

$$q^\star = \frac{q\,\beta}{Z}, \qquad Z = \mathbb{E}_q[\beta] \qquad\qquad d = F\nabla_\mu \log Z = \mu^\star - \mu$$

q: reference belief q*: tilted distribution μ*: selected target moments
β: label-conditioned likelihood message, fixed during the local update.

Exact normalizer: exact local target. Approximate normalizer: corresponding approximate correction.



μ = (m, s): mean and raw second moment d: moment correction θ: weights and biases

Module rule: one child contribution
Sum shared uses and direct terms.

$$d_{p\leftarrow c} = F_p\, J_{c,p}^{\mathsf{T}}\, F_c^{-1} d_c$$

μ*: local target. q⁺: persistent update with the fixed prior, EP schedule, and damping.

Learning objectives and retained correlations

The same module interface supports two independent choices.

How beliefs are learned

EP matches a local tilted target.
MFVI maximizes expected log likelihood minus KL to the prior.

MFVI update (Proposition C.1)

$$g^{\mathrm{VI}} = \nabla_\mu \Psi + \eta_0 - \eta$$

$$d^{\mathrm{VI}} = F g^{\mathrm{VI}}$$ moment-space direction

$$\eta^+ = \eta + \rho\, g^{\mathrm{VI}}, \qquad \mu^+ = \nabla A(\eta^+)$$

Ψ: expected log likelihood
η, η₀: current and prior natural parameters
ρ: step size ∇A: natural parameters to moments

The likelihood derivative backpropagates.
Prior and entropy contribute once.

Which dependencies are retained

Diagonal → **Covariance blocks**

V1	0	0	0
0	V2	0	0
0	0	V3	0
0	0	0	V4

V1	C12	0	0
C12	V2	0	0
0	0	V3	C34
0	0	C34	V4

v: variance c: covariance

Activation and weight correlations use more memory and arithmetic.

$$C_z = \mathbb{E}_W[W C_a W^{\mathsf{T}}] + \mathrm{Cov}_W(W m_a) + C_b$$

C: covariance m_a: input mean
z = Wa + b: weights W, input a, bias b
W, a, b are independent blocks.

Available with either EP or MFVI.

Uncertainty quality and computation

Bayesify (EP): WRN-28-10, CIFAR-10 ID / CIFAR-100 OOD

Method	NLL ↓	ECE % ↓	OOD AUROC ↑	Training-step time ↓	Prediction time ↓
Deterministic[1]	0.205	2.5	0.78	1.00×	1.00×
Ensemble (5)	0.180	1.5	0.88	≈5×[2]	4.77×
MC dropout (20)	0.190	2.0	0.82	≈1×[2]	18.16×
Bayesify (EP)	**0.186**	**1.7**	**0.87**	**3.72×**	**3.29×**

Quality: three-seed means. Timing: relative to deterministic, same hardware, prediction batch size 1.

[1] Quality uses temperature scaling. [2] Training estimates: five serial ensemble updates or one dropout update. Ensemble prediction is serial. Training-step time is not total-training time.

Paper

Bayesify learns beliefs with forward–backward computation and predicts uncertainty in one moment pass.