Bayesian Backprop as Belief Propagation: Single-Pass Predictive Uncertainty

Wen Dong
Air Force Research Laboratory
wen.dong@afrl.af.mil

Abstract

Modern ML systems increasingly use uncertainty to decide when to abstain, route to a fallback, retrieve more context, or spend additional computation. Yet many strong Bayesian deep-learning recipes make uncertainty expensive on the serving path: ensembles, posterior samples, and MC dropout require repeated forward passes, while fast deterministic confidence heads are not generally posterior-predictive computations over parameter beliefs. We introduce *Bayesify*, a message-passing view that turns a deterministic neural computation graph into a single-pass posterior-predictive moment graph. Trainable tensors become Gaussian beliefs, activations carry selected moments, and prediction is one deterministic forward propagation of those moments. Theoretically, we identify exact junction-tree message passing with pseudo-gradient backpropagation in mean space: adding the backward correction to the forward moments gives posterior moments in one update. Reverse-mode adjoints and local Fisher geometry also generate EP and mean-field variational belief updates. This yields a practical wrapper for dense, convolutional, residual, and adapter modules. Experiments evaluate calibrated uncertainty, selective prediction, and OOD behavior using one predictive moment pass, comparing against deterministic calibration, ensembles, dropout, Laplace, SWAG, and variational baselines.

1 Introduction

Uncertainty is most useful when it changes a decision. Deployed ML increasingly looks like a system of modules rather than a single classifier: pretrained models with adapters, retrieval and tool-use pipelines, perception stacks, and fallback policies. In such systems, a predictive distribution is a serving-time control signal. It can decide whether to abstain, route to a larger model, retrieve more context, request human review, or spend more test-time computation. This makes uncertainty an interface requirement: it should be calibrated, local to the module making the decision, and cheap enough to compute on every request. The need is especially clear under distribution shift, where modern networks can be over-confident even when their accuracy degrades [8, 32], and in multi-module pipelines where unreliable hand-offs create hidden technical debt [35, 44].

Current uncertainty methods force a tradeoff between statistical quality and serving cost. Deep ensembles, SWAG, MC dropout, and posterior sampling often improve calibration and robustness, but they require multiple model evaluations or stochastic samples at inference [20, 26]. Mean-field Bayesian neural networks are trainable, but often underestimate epistemic uncertainty and commonly rely on test-time sampling [3]. Laplace and K-FAC approximations retrofit local Gaussian posteriors around a trained solution, but they are usually post-hoc and do not by themselves define a compositional posterior-predictive computation through arbitrary modules [4, 14, 33]. Deterministic uncertainty heads are fast, but they are not generally Bayesian computations over parameter beliefs [27, 39]. The gap is therefore not simply better ECE on a held-out set. The gap is Bayesian uncertainty that can be served in one pass and composed through the same graph as the model.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We address this gap with **Bayesify**: a wrapper that views a neural network as a moment graph. A Bayesified module replaces a point parameter tensor by a Gaussian belief and replaces a point activation by selected moments, usually mean and variance. Dense, convolutional, residual, concatenation, and adapter modules propagate these moments through the same deterministic graph structure as the original model. During EP training, the observation head combines the label with the current output moments to form target output moments; the backward pass carries the resulting correction to the parameter beliefs. We use "single-pass" to refer to inference: the predictive distribution is computed by one deterministic pass of moment propagation, without MC samples or ensembles. Training still follows the ordinary forward–backward minibatch workflow, except that the trainable state is a posterior belief rather than a point tensor.

The theory explains why this forward–backward computation is Bayesian message passing rather than an ad hoc variance heuristic. On exact junction trees, backpropagated pseudo-gradients are posterior-minus-reference moment differences, so adding them gives exact posterior moments in one update. Neural networks require projected statistics, so we prove an operational EP result: reverse-mode adjoints through the chosen moment graph become projected EP moment updates after a local Fisher conversion. In the Gaussian case this gives explicit first- and second-moment increments; in linear-Gaussian chains it reduces to Kalman prediction and Rauch–Tung–Striebel smoothing. The same covariance-matching view also separates our EP-style update from mean-field or Laplace-style precision updates. Appendix C derives MFVI updates; Appendix B gives Gaussian moment rules retaining activation and weight correlations within blocks.

Our contributions are: (i) an information-geometric formulation computing natural-gradient directions by junction-tree message passing and exact posterior moments by a unit mirror step; (ii) a projected-EP theorem that turns reverse-mode adjoints through a neural moment graph into local Bayesian moment updates; (iii) a Bayesify wrapper that turns deterministic modules, including residual and adapter blocks, into single-pass uncertainty modules; and (iv) experiments on calibration, selective prediction, shift, and OOD detection that compare one-pass predictive uncertainty with multi-pass and post-hoc Bayesian baselines. Sec. 2 introduces the moment-graph notation, Sec. 3 gives the theory and algorithm, Sec. 4 reports experiments, and Sec. 5 discusses related work. Across experiments, we therefore emphasize not only ECE/NLL but also selective prediction and equivalent forward passes, because the method is intended to make uncertainty actionable at serving time.

2 Notation & Preliminaries

We view a neural network as a directed graph of local probabilistic modules. A module receives activation tensors from its parents, applies a local factor such as an affine map and a fixed nonlinearity, and emits another activation tensor. In the MLP shown in Fig. 1, $C_\ell = \{x_{\ell-1}, W_\ell, b_\ell, x_\ell\}$ denotes a layer clique, and $S_\ell = \{x_\ell\}$ is the activation vector shared by adjacent cliques. Local consistency means that adjacent modules agree on the retained moments of their shared activation; for exact beliefs, their separator marginals agree. In residual or densely connected networks, one activation may feed several modules, and it keeps a single shared moment state. This is the graphical-model picture used throughout the paper: modules send moment messages forward, adjoints flow backward, and a weight block is updated from the adjoints accumulated at all its uses. We use standard message-passing language from graphical models and variational inference [19, 29, 40, 46].

Each module computes selected output moments from the moments of its inputs and parameters. For independent scalars w, a, for example, $\mathbb{E}[wa] = \mathbb{E}[w]\mathbb{E}[a]$ and $\mathbb{E}[(wa)^2] = \mathbb{E}[w^2]\mathbb{E}[a^2]$. These relations define a moment map in place of the deterministic value calculation $z = wa$.

For an uncertain block u, let $q_u(z; \eta_u)$ be an exponential-family belief with selected sufficient statistics T_u, canonical parameter η_u, mean parameter $\mu_u = \mathbb{E}_{q_u}[T_u(z)]$, and local Fisher block $F_u = \partial\mu_u/\partial\eta_u = \mathrm{Cov}_{q_u}[T_u(z)]$. The algorithm works in mean-parameter space. Each primitive r in the network defines a differentiable projected moment map, written compactly as $\mu_c = \mathcal{M}_r(\{\mu_p : p \in \mathrm{pa}(r)\})$, from parent moments to child moments. In Fig. 1, the output moments and label define the local EP log normalizer $\ell = \log Z_y$. Reverse mode gives ordinary moment derivatives $g_u = \nabla_{\mu_u}\ell$. For a fixed likelihood message, the local Fisher map gives the finite tilted-minus-reference correction $d_u = \Delta\mu_u = F_u g_u$.

In the neural implementation, most uncertain blocks are diagonal Gaussian: $q_u = \mathcal{N}(m_u, \mathrm{Diag}\, v_u)$. We store mean and variance (m_u, v_u), but the Gaussian mean parameters are (m_u, s_u) with

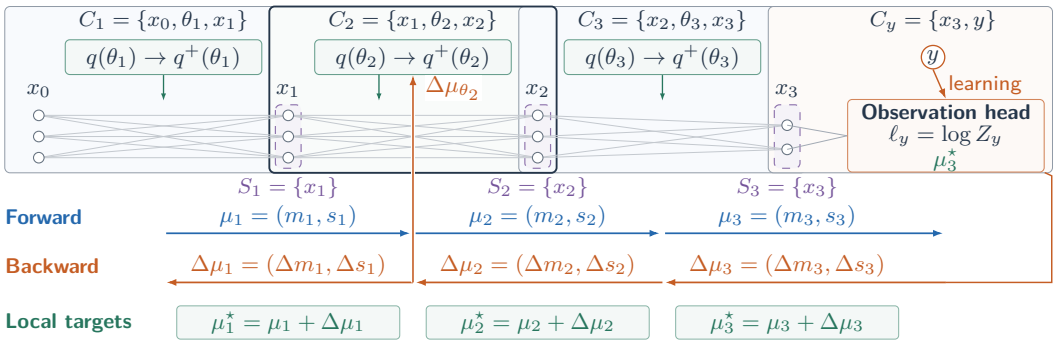


Figure 1: **Bayesify on one input through three untied layers.** Layer cliques C_ℓ share activation separators S_ℓ, and parameter beliefs are defined over $\theta_\ell = (W_\ell, b_\ell)$. Blue arrows carry means and raw second moments $\mu_\ell = (m_\ell, s_\ell)$; orange arrows carry EP corrections $\Delta\mu_\ell = (\Delta m_\ell, \Delta s_\ell)$. Stars denote local activation targets under the chosen approximation; q^+ denotes the complete parameter-belief update, including the fixed prior and damping. Prediction uses only forward moments and probability readout.

$s_u = m_u^{\odot 2} + v_u$. This distinction matters because the EP/natural-gradient step is a moment update, not an ordinary Euclidean step on the stored variance. In the exact Gaussian examples used for checks and proofs, the diagonal variance is replaced by a covariance Σ_u, and the second mean parameter is $\mathbb{E}[uu^\top] = \Sigma_u + m_u m_u^\top$.

A junction tree is a tree of cliques in which the cliques containing any variable form a connected subtree. Consistent clique and separator beliefs then define a joint distribution whose entropy is the sum of clique entropies minus the sum of separator entropies. The tree Bethe objective used below is the expected negative log joint density minus this entropy.

Exact belief propagation combines incoming messages with a local factor and marginalizes to the shared separator. Neural-network factors rarely preserve exact tractable families, so we use projected messages: after the local marginalization or moment calculation, we retain only the selected statistics. This is the expectation-propagation viewpoint: marginalize first, then moment-match. The resulting update is local. Once the reverse sweep has accumulated the adjoint for a block u, the update of q_u uses only its current belief, its local Fisher geometry, and that adjoint. In particular, a weight posterior $q(W_\ell)$ is updated from the adjoints of the layer or adapter in which W_ℓ appears; no global Hessian or full-network posterior covariance is required.

The distinction between projected EP and mean-field VI is especially clear in Gaussian models. If the exact posterior is $\mathcal{N}(\mu, \Sigma)$, projected EP onto diagonal Gaussian beliefs keeps the posterior mean and the diagonal of the covariance, while mean-field VI keeps the mean and the inverse diagonal of the precision Σ^{-1}. This covariance-versus-precision diagnostic is useful below when we distinguish moment-matched updates from faster curvature approximations.

See Appendix E for derivations, classic worked examples, and further background on convex conjugacy, variational inference, and Bethe-style variational principles.

3 Message-Passing Bayesian Backpropagation

With the moment graph defined, we turn to learning its parameter beliefs. Bayesian message updates can be carried backward as pseudo-gradients: update signals expressed in the coordinates of the retained beliefs. Bayesify generates each reverse rule by differentiating the module's forward moment equations and converting its adjoints into belief corrections. Table 1 compares this computation with ordinary backpropagation.

Let U index the parameter tensors selected for Bayesification, with all remaining parameters held at point values. The target posterior and its diagonal approximation are

$$p(W_U \mid D) \propto p_0(W_U) \prod_{n=1}^{N} p(y_n \mid x_n, W_U), \qquad q(W_U) = \prod_{u \in U} \mathcal{N}(M_u, \operatorname{Diag} V_u).$$

Table 1: Ordinary neural propagation and the EP instantiation of Bayesify.

	Ordinary neural network	Bayesified moment network
Trainable state	Point parameter tensors	Gaussian parameter beliefs
Forward pass	Activation values	Activation means and variances
Backward pass	Activation and weight derivatives	Moment adjoints and local belief corrections
Parameter update	Update point tensors	Update Gaussian parameter beliefs
Prediction	One forward evaluation	One forward moment propagation

Parameter beliefs persist across minibatches; activation moments are recomputed for each input. We establish the backward correspondence on exact junction trees, derive the neural EP algorithm, and then extend its rules to MFVI and correlated beliefs.

Exact reference case. In a tree-structured graphical model, clique and separator beliefs can be represented by exponential-family mean parameters. Standard belief propagation enforces local consistency: each clique marginalizes to its separator, and neighboring cliques agree on the separator belief. On a junction tree, this procedure is exact. The following statement is informal; Appendix A gives the formal exact-tree derivation and worked Gaussian examples.

Exactness requires that the clique and separator families represent the relevant message products and marginals, including their local variations. We use proper normalized messages, regular minimal coordinates, and interior beliefs so that the local Fisher maps are invertible.

Theorem 3.1 (Informal: exact inference by pseudo-gradient backpropagation)**.** *Run an exact normalized junction-tree collect pass, retaining its log normalizers. For a cached clique or separator belief, let g_u be the reverse derivative of the log evidence with respect to its moments μ_u, holding its cached log scale fixed. The pseudo-gradient $d_u = F_u g_u$ obeys*

$$\begin{aligned} d_u &= F_u g_u^{\mathrm{loc}} + \sum_c F_u J_{c,u}^\top F_c^{-1} d_c, \\ \mu_u^{\mathrm{post}} &= \mu_u + d_u. \end{aligned} \tag{1}$$

Here $J_{c,u} = \partial \mu_c / \partial \mu_u$, and g_u^{loc} includes direct local-normalizer derivatives. One forward sweep and one reverse sweep give exact posterior clique and separator moments.

Proof sketch. At a message cut, the remaining factors give a fixed likelihood β_u and a posterior proportional to $q_u \beta_u$. Differentiation gives $\nabla_{\eta_u} \log Z = \mu_u^{\mathrm{post}} - \mu_u = F_u g_u$. The chain rule through exact message products, marginals, and log normalizers gives the recursion. Corrected beliefs agree on shared separators. For a prior input, d_u is posterior minus prior moments. Appendix A gives the full proof and mirror-update interpretation.

Projected neural moment graphs. Neural networks do not generally permit exact junction-tree inference with tractable clique marginals. There are two sources of approximation. First, local neural factors are often outside the chosen tractable family: Wx is bilinear when both W and x are uncertain, and nonlinear activations do not usually preserve Gaussianity. Second, the computational graph can have high treewidth or loopy structure because of residual branches, DenseNet concatenations, attention, weight sharing, or repeated use of the same activation. Exact inference would require large cliques that are computationally infeasible.

We therefore work with a projected moment graph. Each primitive computes selected output statistics from parent statistics, exactly when available and approximately otherwise. This is the expectation-propagation viewpoint: combine local information, marginalize or compute predictive moments, and then project to the retained statistics. In diagonal Gaussian implementations, the retained statistics are typically means and diagonal variances. Skip, residual, and concatenation operations are treated as shared-variable moment primitives with one source moment state.

The next theorem is the operational bridge from the exact message-passing theorem to the neural algorithm. For loopy or high-treewidth neural graphs, it describes the scheduled projected-EP update associated with the chosen moment graph. The statement distinguishes an exact tilted-moment calculation from the correction obtained with an approximate normalizer.

Theorem 3.2 (Informal: projected-statistic EP by adjoint natural gradients)**.** *Consider a differentiable moment graph whose uncertain blocks u are represented by tractable exponential-family beliefs $q_u(z;\eta_u)$ with selected statistics T_u, mean parameters $\mu_u = \mathbb{E}[T_u]$, and Fisher blocks $F_u = \partial\mu_u/\partial\eta_u$. Suppose each primitive computes an exact or declared approximate projected moment map from parent moments to child moments. For block u, let $\ell_u = \log Z_u$ be the log normalizer obtained by combining its reference belief with a fixed downstream likelihood message. If the moment computation evaluates this normalizer, reverse-mode differentiation gives $g_u = \nabla_{\mu_u}\ell_u$, and*

$$\mu_u^\star = \mu_u + F_u g_u \tag{2}$$

is the tilted moment target relative to that reference. An approximate normalizer gives its corresponding approximate correction. A damped update is $\mu_u^+ = (1-\rho)\mu_u + \rho\mu_u^\star$ when the target and interpolation are admissible.

Proof sketch. Let the descendants of block u, together with the observation head, induce a downstream likelihood message $\beta_u(z)$. Define the local normalizer $Z(\eta_u) = \int \beta_u(z)q_u(z;\eta_u)\,dz$, and let the tilted distribution be proportional to $\beta_u(z)q_u(z;\eta_u)$. Differentiating $\log Z$ with respect to the canonical parameter gives the difference between tilted and reference moments, $\nabla_{\eta_u}\log Z = \mathbb{E}_{\text{tilt}}[T_u] - \mathbb{E}_q[T_u]$. Since $\nabla_{\eta_u}\log Z = F_u\nabla_{\mu_u}\log Z$, the tilted moment is $\mu_u + F_u\nabla_{\mu_u}\log Z$. Reverse mode supplies this derivative when the composed moment computation evaluates the stated normalizer.

Theorem 3.2 converts moment-graph adjoints into EP belief corrections. The forward pass computes projected moments. The observation model supplies the terminal correction. The backward pass transports its adjoints through the retained moments. The local Fisher block converts those adjoints into EP moment corrections. For exact linear-Gaussian normalizers, these targets match Gaussian belief propagation or Kalman smoothing. With diagonal statistics, nonlinear moment approximations, and finite scheduled updates, it gives a projected EP approximation. Appendix B gives the formal projected-EP statement, the diagonal Gaussian message conversions used below, and the shared-variable results for residual and concatenation primitives.

A forward moment map specifies how uncertainty is propagated, but does not determine how the observed data should update weight and activation beliefs. EP forms tilted distributions and matches their moments; MFVI optimizes a variational objective. They can share the same module derivatives while using different terminal signals and parameter-belief updates. We now develop the EP construction for diagonal Gaussian beliefs.

Diagonal Gaussian moment messages. For $q(u) = \mathcal{N}(m_u, \operatorname{Diag} v_u)$, we store (m_u, v_u) but update raw moments (m_u, s_u), $s_u = m_u^{\odot 2} + v_u$. The backward correction $\Delta u = (\Delta m_u, \Delta s_u)$ points toward the tilted moments. With stored-coordinate adjoints $g_m = \partial\ell/\partial m$ and $g_v = \partial\ell/\partial v$, Theorem 3.2 gives

$$\Delta m = v \odot g_m, \qquad \Delta s = 2m \odot v \odot g_m + 2v^{\odot 2} \odot g_v. \tag{3}$$

Equivalently, the tilted moments are $m^\star = m + \Delta m$ and $s^\star = s + \Delta s$, with $v^\star = s^\star - (m^\star)^{\odot 2}$. If the implementation minimizes $\mathcal{L} = -\ell$, the adjoints in Eq. (3) are replaced by $-\partial\mathcal{L}/\partial m$ and $-\partial\mathcal{L}/\partial v$.

A module that receives a child increment needs the reverse conversion. For a child state (m, v) and incoming increment $(\Delta m, \Delta s)$, the adjoints with respect to the stored coordinates are

$$g_m = \Delta m \oslash v, \qquad g_v = (\Delta s - 2m \odot \Delta m) \oslash (2v^{\odot 2}), \tag{4}$$

where $\oslash$ denotes elementwise division. Equations (3) and (4) therefore encode adjoints as moment corrections and decode incoming corrections for differentiation through a module.

For a persistent belief, first form its complete target under the declared EP schedule. With $\Delta m = m^\star - m$ and $\Delta s = s^\star - s$ measured from that persistent belief, raw-moment damping uses $\rho \in [0, 1]$:

$$m^+ = m + \rho\Delta m, \qquad s^+ = s + \rho\Delta s, \qquad v^+ = s^+ - (m^+)^{\odot 2}. \tag{5}$$

This is a convex combination in moment space, $(m^+, s^+) = (1-\rho)(m, s) + \rho(m^\star, s^\star)$. If the current and target moments are both valid, their convex interpolation remains in the Gaussian moment domain. Approximate targets require a positivity safeguard; the implementation also enforces a small variance floor. The quantity returned by a Bayesified backward pass is therefore a local moment increment for a posterior belief, rather than a Euclidean gradient of a point weight.

Bayesified module rules. A Bayesified module maps input moments and local parameter beliefs to output moments. For an affine module $z = Wa + b$, write $M = \mathbb{E}[W]$ and let V_W collect its entrywise variances. Its forward rule is

$$m_z = Mm_a + m_b, \qquad v_z = M^{\odot 2} v_a + V_W(m_a^{\odot 2} + v_a) + v_b, \tag{6}$$

where $q(W) = \mathcal{N}(M, \operatorname{Diag} V_W)$. The input, weight, and bias blocks are independent, and the retained input and weight covariances are diagonal. Convolution, embedding, and adapter projections use the same deterministic operator as the original module, applied to the corresponding mean and second-moment terms. The variance formula is the law of total variance: input uncertainty is propagated through mean weights, and weight uncertainty acts on the input second moment.

Although Wa is generally non-Gaussian, Eq. (6) gives its exact means and marginal variances under these independence assumptions. Representing the output by a Gaussian is the projection; the backward rule differentiates this moment calculation.

The backward rule receives a child moment increment and returns parent moment increments together with local moment increments for any persistent parameter beliefs in the module. It first converts the child increment into child adjoints by Eq. (4), differentiates the cached forward moment computation such as Eq. (6), and then converts the resulting parent and parameter adjoints into moment increments by Eq. (3). This is the module-level analogue of ordinary backpropagation: a module sends messages to its parents and returns a local update direction for its own trainable belief.

For scalar $z = wa + b$, let r and t be the adjoints of m_z and v_z, obtained from the incoming correction by Eq. (4). Differentiating Eq. (6) gives

$$g_{m_w} = m_a r + 2m_w v_a t, \qquad g_{v_w} = (m_a^2 + v_a)t.$$

Equation (3) gives the weight correction relative to its cached reference. Once the schedule forms the persistent target, Eq. (5) gives its raw-moment damping. The same differentiation returns the input correction to the preceding module.

From the label to an output correction. The classification head conditions uncertain logits $q(z) = \mathcal{N}(m, V)$, $V = \operatorname{Diag}(v)$, on the label target y. Cache $\bar{m} = m$, $\bar{p} = \operatorname{softmax}(\bar{m})$, and $\bar{J} = \operatorname{Diag}(\bar{p}) - \bar{p}\bar{p}^\top$. Hold these quantities fixed in the local model $y = \bar{p} + \bar{J}(z - \bar{m}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, RI)$ and $R > 0$ controls observation noise. At the cached state, Gaussian conditioning gives

$$K = V\bar{J}^\top(\bar{J}V\bar{J}^\top + RI)^{-1}, \qquad m^\star = m + K(y - \bar{p}), \qquad V^\star = V - K\bar{J}V.$$

The head retains class coupling until it sends $\Delta m = m^\star - m$ and $\Delta s = \operatorname{diag}(V^\star) + (m^\star)^{\odot 2} - (v + m^{\odot 2})$ backward. These moments are exact for the local observation model; the softmax likelihood has been approximated. The monitored classification loss is separate from this correction. Prediction omits label conditioning and maps output moments to probabilities.

Maintaining the prior during learning. Explicit-site EP retains the fixed prior when removing a likelihood site. In tied power EP, it enters the shared cavity and its normalization (Appendix B.1). MFVI includes the prior once through its KL term (Appendix C). The learning schedule specifies the reference and finite update.

Algorithm 1 retains the forward–backward training schedule, with parameter beliefs as persistent state and moment corrections as backward messages. The BAYESIFY wrapper replaces tensors by Gaussian beliefs and deterministic operations by moment operations, retaining shared inputs at residual additions and concatenations. Dense, convolutional, embedding, and adapter modules reuse their deterministic operators on means and second moments; other differentiable modules use declared analytic or first-order projections. TensorFlow or PyTorch custom VJPs implement the reverse rules; the EP schedule completes parameter updates, optionally using raw-moment damping (Eq. (5)).

Table 2 summarizes the forward and backward module rules.

Table 2 treats graph operations as part of the same moment computation. Reused activations share one moment state, and backward contributions are summed there, as in ordinary autodiff. For $y = Sx + F(x, W)$, the composite sensitivity $S + J_x$ retains the shared input's contribution. The skip path participates in uncertainty propagation without acquiring a separate posterior.

Algorithm 1 Bayesified forward–backward pass

Require: Bayesified graph; Gaussian beliefs and fixed priors; EP reference/update schedule; input x; label y during training.

For training, form the schedule's parameter references; for prediction, use the persistent beliefs.
Initialize input moments, usually $m_x = x$ and $v_x = 0$.
In forward order, compute each module's child moments and cache its moment calculation.
For prediction, return the output-moment readout and stop.
For training, condition the head on y to obtain $(\Delta m, \Delta s)$.
for each Bayesified module in reverse order **do**
    Convert the child increment to child adjoints using Eq. (4).
    Differentiate the cached forward moment computation.
    Encode parent/parameter corrections by Eq. (3), summing shared uses.
end for
Complete the declared EP update of each persistent belief, retaining its fixed prior (Appendix B.1).

Table 2: **Bayesified module rules.** Reverse rules differentiate the declared forward moments between the conversions in Eqs. (4) and (3).

Module	Forward moment computation	Backward return
Dense, Conv, embedding, adapter	Eq. (6), using the module's deterministic operator on mean and second-moment terms.	Corrections to input moments and local weight/bias beliefs.
Nonlinearity	Analytic Gaussian $m_h = \mathbb{E}[\phi(Z)]$, $v_h = \mathrm{Var}[\phi(Z)]$ when available; otherwise a declared first-order projection.	Input correction; parameter correction only for trainable nonlinearities.
Residual or skip	For $y = Sx + F(x, W)$, reuse x and retain the diagonal of $(S + J_x)V_x(S + J_x)^\top + J_w \Sigma_w J_w^\top + V_\epsilon$ under independent input, weight, and noise blocks. Here J_x, J_w are branch sensitivities and Σ_w is weight covariance.	Sum shared-input corrections; update residual-branch parameter beliefs.
Concatenation	Concatenate means; carry shared-input cross-covariances into a fused next module or project them away explicitly.	Return corrections to every source, summing shared uses.
Observation head	Condition on the label under the declared observation model; classification uses the local Gaussianized head.	Initial output moment correction.

The linear-Gaussian case provides a compact check on the algorithm. When a sequence of modules is represented by checkpoint-local linear-Gaussian transition factors and retains sufficient Gaussian moments, the resulting probability model is a Kalman chain. The Bayesified forward pass is Kalman prediction, and the backward moment increments recover the Rauch–Tung–Striebel smoothing update.

Variational learning with the same reverse construction. For MFVI, let $q(W) = \prod_u q_u(W_u)$ and $\Psi = \mathbb{E}_q[\log p(D \mid W)]$. The reverse sweep computes $\nabla_{\mu_u} \Psi$; subtracting the local KL derivative once gives

$$g_u^{\mathrm{VI}} = \nabla_{\mu_u} \Psi + \eta_{0,u} - \eta_u, \qquad d_u^{\mathrm{VI}} = F_u g_u^{\mathrm{VI}},$$
$$\eta_u^+ = (1 - \rho)\eta_u + \rho\big(\eta_{0,u} + \nabla_{\mu_u} \Psi\big).$$

Here $\eta_{0,u}$ is the prior's canonical parameter. The pseudo-gradient d_u^{VI} is the ELBO's natural-gradient direction in mean coordinates; the canonical mirror step [16] must remain in the valid parameter domain. Moment-damped EP uses a tilted target; the displayed MFVI step uses a canonical target. Exact expectations give the ELBO update; approximate moments give the corresponding surrogate. Appendix C derives the Gaussian forms.

Correlations among activations and weights. Diagonal variances can be replaced by covariance blocks. For $z = Wa + b$ with independent blocks W, a, b, the law of total covariance gives

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

The first term propagates activation covariance through uncertain weights; the second propagates weight uncertainty acting on the mean activation. Both within-activation and within-weight correlations are retained. Differentiating this map and applying the full Gaussian Fisher conversion generates the reverse rules (Appendix B.6). Weight–input dependence requires additional joint moments.

Approximation scope. The posterior family, nonlinear closures, retained correlations, and update schedule determine the approximation. Discarding input correlations omits $2\sum_{j<k} M_{ij}M_{ik}(C_a)_{jk}$ from an affine output variance even with fixed weights; later modules can amplify this error. Same-belief sampling assesses propagation error, separately from posterior-approximation quality. The local identities do not guarantee convergence of the outer neural training loop.

Bayesification adds the computation needed to propagate and update the retained moments. Learning requires a forward moment pass, its reverse pseudo-gradient calculation, and local parameter-belief updates; prediction requires only the forward moment pass. Diagonal Gaussian beliefs store a mean and variance for each Bayesian parameter, while affine and convolutional modules evaluate the mean operation together with the variance terms in Eq. (6). Retaining covariance blocks increases computation and storage; diagonal projections and nonlinear moment closures reduce that cost by discarding information that can affect calibration, OOD detection, and selective prediction.

4 Experiments

This section evaluates our EP instantiation of Bayesify on standard vision and NLP benchmarks, comparing calibration, out-of-distribution (OOD) detection, and inference efficiency to ensembles, SWAG, Laplace, and single-pass baselines. We report results on CIFAR-10/C, Camelyon17-WILDS, and CLINC OOS, and study parameter-efficient fine-tuning with Bayesian LoRA adapters on the `gpt2_base_en` backbone (SST-2, RTE with OOD IMDb/HANS). Implementation details, training recipes, and additional regression/visualization results are provided in Sec. D.

4.1 Performance, Calibration, and Efficiency

We report accuracy and NLL for predictive quality; ECE and Brier-style diagnostics for calibration; entropy-based AUROC for OOD separation; and risk–coverage or AURC for selective prediction, where high-uncertainty examples are withheld. Predictive-pass counts distinguish one moment-graph traversal from repeated model evaluations; measured runtime ratios quantify the actual cost.

For vision, CIFAR-10 measures ID classification, CIFAR-10-C measures corruption robustness (cNLL/cECE across severities) [9], and CIFAR-100 supplies near-OOD inputs for AUROC. Camelyon17-WILDS tests generalization across hospitals [18]. CLINC OOS evaluates intent classification and out-of-scope detection [21]. Architectures are WRN-28-10 for CIFAR, DenseNet-121 for Camelyon17, and RoBERTa-base for CLINC. Data preparation and training details are in Appendix D.

Baselines span common uncertainty families: deterministic networks with temperature scaling (Det+TS) [8]; deep ensembles [20]; SWAG [26]; last-layer Laplace with diagonal/K-FAC precision [4, 34]; single-pass methods SNGP and DUE [23, 38]; MC-dropout [5]; and Bayes-by-Backprop (BBB) [3]. Metrics include ID accuracy, NLL, ECE; OOD AUROC (entropy score); and corruption metrics cNLL/cECE on CIFAR-10-C.

Temperature scaling is included because calibration alone can improve a deterministic confidence score without learning parameter uncertainty. Its temperature is fitted once on the ID development set and held fixed under shift and OOD evaluation.

Table 3 shows that our EP method matches or closely approaches ensembles and SWAG on calibration and robustness while preserving *single-pass* inference. On CIFAR-10-C, EP cuts cNLL by 20.4% versus Det+TS and is within a few percent of a 5-member ensemble; see the CIFAR-10 block (cNLL/cECE columns). For OOD, EP improves AUROC over Det+TS ($\approx$0.09 on CIFAR-100; 0.06 on CLINC OOS) and is competitive with sampling-based methods. On Camelyon17-WILDS, EP maintains strong ID/ worst-domain NLL/ECE without multi-pass inference. These results show competitive calibration and OOD detection with one predictive moment pass.

The tables report three-seed means. Table 4 compares measured WRN costs: Bayesify takes $3.29\times$ the deterministic prediction time, versus $4.77\times$ for a serial five-member ensemble and $18.16\times$ for

Table 3: Calibration and out-of-distribution evaluation across vision and NLP tasks. Entries are means of three seeds.

Metric	Det+TS	Ens (5)	SWAG (20)	Laplace (20)	SNGP/ DUE	MC-dropout	BBB	Ours (EP)
CIFAR-10 (WRN-28-10): ID = CIFAR-10; OOD = CIFAR-10-C/C100								
ID Acc.↑	95.1	95.8	95.7	95.5	95.3	95.4	95.2	**95.6**
ID NLL↓	0.205	0.180	0.184	0.192	0.196	0.190	0.188	**0.186**
ID ECE↓	0.025	0.015	0.018	0.021	0.022	0.020	0.019	**0.017**
OOD AUROC↑	0.78	0.88	0.86	0.84	0.83	0.82	0.83	**0.87**
OOD cNLL↓	0.685	0.540	0.560	0.590	0.610	0.580	0.575	**0.545**
OOD cECE↓	0.078	0.044	0.049	0.056	0.060	0.055	0.054	**0.046**
Camelyon17-WILDS (DenseNet-121): ID = Avg-domain; OOD = Worst-domain								
ID Acc.↑	93.8	94.5	94.2	94.1	93.9	94.0	93.9	**94.4**
ID NLL↓	0.210	0.162	0.170	0.178	0.185	0.180	0.179	**0.165**
ID ECE↓	0.030	0.018	0.021	0.025	0.027	0.026	0.025	**0.017**
OOD AUROC↑	0.74	0.80	0.79	0.78	0.77	0.78	0.78	**0.80**
OOD NLL↓	0.315	0.260	0.275	0.290	0.298	0.285	0.282	**0.262**
OOD ECE↓	0.095	0.060	0.065	0.072	0.078	0.070	0.068	**0.061**
CLINC Intent (RoBERTa-base): ID = In-scope; OOD = Out-of-scope								
ID Acc.↑	90.4	91.0	90.8	90.7	90.6	90.8	90.7	**91.1**
ID NLL↓	0.42	0.38	0.39	0.40	0.41	0.39	0.39	**0.38**
ID ECE↓	0.045	0.030	0.032	0.035	0.038	0.033	0.034	**0.028**
OOD AUROC↑	0.86	0.92	0.91	0.90	0.89	0.90	0.90	**0.92**
OOD NLL↓	0.95	0.81	0.85	0.88	0.89	0.85	0.84	**0.80**
OOD ECE↓	0.110	0.065	0.075	0.080	0.085	0.078	0.077	**0.062**

20-sample MC-dropout. Its training-step ratio is $3.72\times$; one moment pass therefore has a measurable constant-factor cost.

4.2 LLMs & PEFT: Bayesian adapters without multi-sampling

We evaluate parameter-efficient Bayesian fine-tuning of the `gpt2_base_en` language-model backbone using LoRA adapters [13] on two GLUE [41] in-distribution (ID) tasks—sentiment classification (SST-2) and textual entailment (RTE)—along with domain-mismatched out-of-distribution (OOD) datasets: IMDb reviews [24] (OOD relative to SST-2) and the adversarial NLI dataset HANS [28] (OOD relative to RTE). We compare deterministic LoRA with temperature scaling [8], last-layer Laplace approximation [4, 34], mean-field variational inference (VI) [3], and our EP approach (single-pass, no test-time sampling). Metrics include ID accuracy, calibration (NLL, ECE averaged across tasks), OOD detection via predictive entropy AUROC, and selective prediction (AURC; 7).

Table 5 reports three-seed means. EP matches Laplace on the two OOD AUROCs and has the lowest NLL, ECE, and AURC among these methods, using one moment pass rather than the 20 and 10 predictive samples used by Laplace and VI. LoRA is applied to GPT-2 attention projections (rank 16, dropout 0.05), with diagonal Gaussian beliefs over both adapter factors. Training uses at most five epochs, with checkpoints selected by development-set NLL. Further details are provided in supplementary materials.

5 Related work

Bayesian neural networks reason with a posterior predictive [2, 25, 30]. Scalable approximations include sampling and model averaging—HMC/SGLD, ensembles, and SWAG [20, 26, 31, 42]; variational methods such as BBB and MC-dropout [3, 5, 12]; and post-hoc Laplace approximations [4, 34]. These commonly require multiple predictive evaluations. Single-pass alternatives include evidential heads and the Gaussian-process-based SNGP and DUE [23, 36, 38]. Calibration under shift and decision-aware evaluation motivate NLL/ECE, OOD AUROC, and risk–coverage metrics [1, 6, 8, 32].

Graphical-model foundations (junction trees; belief propagation) and variational/free-energy views connect message passing to local exponential-family structure [19, 29, 40, 46]. Probabilistic back-

Table 4: WRN-28-10 cost relative to the deterministic model (same hardware; latency at batch size 1). Bayesify's training-step ratio is 3.72; dashes denote unreported values.

Method	Latency	Peak memory
Deterministic	1.00	1.00
SNGP	1.23	1.20
DUE	1.38	1.31
Ens(5, serial)	4.77	–
MC-dropout(20)	18.16	–
Bayesify	3.29	1.93

Table 5: GPT-2 LoRA (three-seed means; shared frozen backbone and adapter budget). ID NLL/ECE average SST-2/RTE; OOD AUROC uses IMDb/HANS.

Metric	Det+TS	Laplace	VI	Bayesify (EP)
SST-2 Acc.↑	94.2	94.5	94.4	**94.6**
RTE Acc.↑	86.0	86.2	86.1	**86.3**
ID NLL↓	0.320	0.290	0.300	**0.280**
ID ECE↓	0.025	0.021	0.022	**0.020**
IMDb AUROC↑	0.82	0.87	0.86	**0.87**
HANS AUROC↑	0.78	0.82	0.81	**0.82**
AURC↓	0.055	0.048	0.050	**0.046**
Predictive passes	1	20	10	1

propagation already differentiates a forward approximate normalizer to learn factorized weight beliefs [10]; feed-forward uncertainty propagation also supports end-to-end moment-based training [37]. Streamlining Prediction propagates a supplied weight posterior, including correlations [22]; PFP combines variationally learned beliefs with compiled Gaussian moment propagation for prediction [17]. Bayesify identifies backward belief corrections as pseudo-gradients and generates their module rules by autodiff and local Fisher conversion, including shared-state operations in residual, concatenation, and adapter graphs. Its theory states when these rules represent Bayesian moment updates. In parameter-efficient fine-tuning, LoRA adapters [13] provide a natural locus for Bayesian treatment; prior adapter-Bayes relies mainly on Laplace/VI at the head/adapters [4, 34]. Our adapter instantiation preserves LoRA's efficiency while improving calibration and selective prediction. Finally, insights on BNN posterior geometry and the cold-posterior effect motivate structured yet tractable approximations and decision-aware evaluation rather than expensive exactness [15, 43].

6 Conclusion

We presented *Bayesify*, a moment-graph view of Bayesian backpropagation that turns deterministic modules into local probabilistic modules. On exact junction trees, a reverse sweep computes posterior-minus-reference moments, giving exact posterior moments in one update. On neural moment graphs, EP converts reverse adjoints into local belief corrections. The resulting algorithm is a standard forward–backward computation with different payloads: the forward pass propagates first and second moments, and the backward pass returns moment increments for local posterior beliefs. This yields a practical single-pass posterior-predictive interface for Dense/Conv layers and adapters, while preserving the graph structure needed for residual and densely connected architectures. Empirically, the method provides calibrated uncertainty and selective/OOD behavior competitive with multi-pass Bayesian baselines, but with one Bayesified predictive pass. The main takeaway for practitioners is simple: uncertainty can be propagated through the same modular computation graph used for ordinary deep learning, provided modules communicate moments rather than only point activations.

Broader impacts

This paper develops a methodological and theoretical connection between message passing on junction trees and natural-gradient optimization of a mean-form variational objective, with the practical goal of enabling single-pass predictive uncertainty in deep networks. Reliable uncertainty estimates can support safer deployment by enabling risk-aware behaviors such as selective prediction/abstention, escalation to human oversight, and detection of distribution shift. However, uncertainty estimates are not a guarantee of correctness or safety and may be miscalibrated under distribution shift or misspecification; using them in high-stakes settings should therefore include application-specific validation, monitoring, and appropriate safeguards. This work does not involve new data collection, human subjects, or the release of sensitive datasets.

Acknowledgments, Disclosure of Funding, and Disclaimer

The author thanks the reviewers for their constructive comments.

The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense or of the United States Air Force. The content or appearance of hyperlinks does not reflect an official DoD, Air Force, Air Force Research Laboratory position or endorsement of the external websites, or the information, products, or services contained therein.

References

[1] Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021.

[2] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[4] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[6] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017.

[7] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*, 2019.

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[9] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[10] Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1861–1869. PMLR, 2015.

[11] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard Turner. Black-box Alpha divergence minimization. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1511–1520. PMLR, 2016. URL `https://proceedings.mlr.press/v48/hernandez-lobatob16.html`.

[12] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. In *JMLR*, volume 14, pages 1303–1347, 2013.

[13] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL `https://api.semanticscholar.org/CorpusID:235458009`.

[14] Alexander Immer, M. Bauer, Vincent Fortuin, Gunnar Rätsch, and Mohammad Emtiyaz Khan. Scalable marginal likelihood estimation for model selection in deep learning. *ArXiv*, abs/2104.04975, 2021. URL `https://api.semanticscholar.org/CorpusID:233210275`.

[15] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.

[16] Mohammad Khan and Wu Lin. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 878–887. PMLR, 2017. URL https://proceedings.mlr.press/v54/khan17a.html.

[17] Bernhard Klein, Falk Selker, Hendrik Borras, Sophie Steger, Franz Pernkopf, and Holger Fröning. Accelerated execution of Bayesian neural networks using a single probabilistic forward pass and code generation, 2025. URL https://arxiv.org/abs/2511.23440.

[18] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, and et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[19] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6405–6416, 2017.

[21] Stefan Larson, Anish Mahendran, and et al. An evaluation dataset for intent classification and out-of-scope prediction. In *EMNLP*, 2019.

[22] Rui Li, Marcus Klasson, Arno Solin, and Martin Trapp. Streamlining prediction in Bayesian deep learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/8d492b8a6201d83d1015af9e264f0bf2-Abstract-Conference.html.

[23] Jeremiah Zhe Liu and et al. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *NeurIPS*, 2020.

[24] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, A. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics*, 2011. URL https://api.semanticscholar.org/CorpusID:1428702.

[25] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[26] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Andrey Malinin and Mark John Francis Gales. Predictive uncertainty estimation via prior networks. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:3580844.

[28] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, 2019.

[29] Thomas Minka. *Expectation Propagation for Approximate Bayesian Inference*. PhD thesis, MIT, 2001.

[30] Radford M Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, University of Toronto, 1995.

[31] Radford M. Neal. Mcmc using hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, chapter 5. CRC Press, 2011.

[32] Yaniv Ovadia, Emily Fertig, Jie Jessie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:174803437.

[33] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

[34] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

[35] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Neural Information Processing Systems*, 2015. URL https://api.semanticscholar.org/CorpusID:17699480.

[36] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

[37] Alexander Shekhovtsov, Boris Flach, and Michal Busta. Feed-forward uncertainty propagation in belief and neural networks, 2018. URL https://arxiv.org/abs/1803.10590.

[38] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression, 2021. URL https://arxiv.org/abs/2102.11409v1.

[39] Joost R. van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, 2020. URL https://api.semanticscholar.org/CorpusID:263889552.

[40] Martin J. Wainwright and Michael I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., 2008.

[41] Alex Wang, Amanpreet Singh, and et al. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2018.

[42] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[43] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świkatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning*, pages 10248–10259, 2020.

[44] Walter F. Wiggins and Ali S. Tejani. On the opportunities and risks of foundation models for natural language processing in radiology. *Radiology. Artificial intelligence*, 4 4:e220119, 2022. URL https://api.semanticscholar.org/CorpusID:250934677.

[45] John Winn and Christopher M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(23):661–694, 2005. URL https://www.jmlr.org/papers/v6/winn05a.html.

[46] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. In *IEEE ISIT*, 2005.

A Exact Junction-Tree Message Passing and Natural Gradients

We model a neural network as a finite directed acyclic graph (DAG) of modules. Each node v receives the stacked activations from its parents, applies an affine, parameterized linear map followed by a fixed, parameter-free nonlinearity, and outputs an activation h_v. Concretely, for parent activations $h_{\mathrm{pa}(v)}$ and node parameters W_v, the preactivation is $a_v = L_v(W_v)\, h_{\mathrm{pa}(v)} + b_v$ (a dense matrix, convolution, or projection), and the node output is $h_v = \phi_v(a_v)$. Optional exogenous noise ξ_v (e.g., dropout masks) may be used inside ϕ_v, but learnable parameters live in the affine map $L_v(W_v)$ (including learned scale/shift used by normalization layers). The global parameter is $W = \{W_v\}_v$. This notation is general and covers standard architectures: dense/conv nets; residual blocks (additive skip connections in the DAG); multi-branch modules (e.g., GLU gates as two affine maps combined by a fixed nonlinearity); and self-attention, where queries/keys/values are produced by affine maps $x \mapsto xW_Q, xW_K, xW_V$, attention weights are computed by a parameter-free softmax, and outputs are projected by an affine W_O. Low-rank adapters (LoRA/PEFT) are also included: they define an effective affine perturbation ΔW applied linearly to the activation; whether ΔW is factorized internally is immaterial at the level of the node map. This modular organization of information flow mirrors how complex systems in biology, engineered networks, and organizations achieve robustness: when the environment changes unexpectedly, one typically adjusts a few modules and their interfaces rather than rewiring the entire network. The multi-layer perceptron notation used in our Notation & Preliminaries section- $a_\ell = W_\ell x_{\ell-1} + b_\ell, x_\ell = f_\ell(a_\ell)$-is the special case where the DAG is a chain (each node has a single parent) and $L_\ell(W_\ell)$ is a dense matrix. We adopt this explicit separation-parameters appear in affine maps, nonlinearities are parameter-free-because it cleanly identifies where parameters enter the computation and will let us attach probabilistic structure to parameter blocks (cliques) while treating inter-node interfaces as separators in the sequel.

We argue that Bayesian inference should be a first-class citizen in our formulation because machine learning aims at **generalization under uncertainty**, not just optimization of an empirical objective. Given our network as a DAG of modules (where parameters only participate in affine maps), let a prior $p(W)$ be fixed independently of an i.i.d. sample D and define, for a loss $\ell \in [0,1]$, the **population risk** of a posterior $q(W)$ as $R_*(q) = \mathbf{E}_{(x,y)\sim P_*}, \mathbf{E}_{W\sim q}, \ell(f_W(x), y)$ and its **empirical counterpart** $\hat{R}_D(q) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{E}_{W\sim q}, \ell(f_W(x_n), y_n)$. For fixed $\lambda > 0$ chosen independently of the sample and $\delta \in (0,1)$, a PAC–Bayesian Hoeffding bound holds with probability at least $1-\delta$, simultaneously for all q with finite $\mathrm{KL}(q\|p)$: $R_*(q) \le \hat{R}_D(q) + \frac{\mathrm{KL}(q\|p)+\ln(1/\delta)}{\lambda} + \frac{\lambda}{8N}$. Thus **choosing and updating a distribution over parameters** — not just a point W — is directly tied to generalization via the divergence $\mathrm{KL}(q|p)$. This perspective also aligns with practice: solutions that are **insensitive to small parameter perturbations** ("wide basins" or flat minima) exhibit better test performance; representing such insensitivity by averaging predictions under a local (q) (e.g., Gaussian around a mode, or dropout-style multiplicative noise viewed as exogenous ξ in our node map) improves generalization. A Bayesian neural network formalizes this averaging as the **posterior predictive** $p(y \mid x, D) = \int p(y \mid x, W) p(W \mid D) dW$, which naturally regularizes training (through the prior) and encodes epistemic uncertainty (through dispersion of $p(W \mid D)$).

At service time, systems require **fast calibrated probabilities** for selective prediction, shift and OOD detection, adversarial robustness, and decision-aware evaluation (e.g., NLL, ECE, selective risk at a target coverage). With our notation the relevant quantities are posterior-predictive functionals of the form $\mathrm{U}(x) = U(p(\cdot \mid x, D))$ (entropy, variance, tail risk), which enable abstention/routing by thresholding $\mathrm{U}(x)$, and **selective risk** defined on the accepted set $\{x : \mathrm{U}(x) \le \tau\}$. Point estimates $f_{\hat{W}}(x)$ can be **systematically over-confident** under distribution shift because they ignore uncertainty in W; posterior-predictive uncertainty directly addresses this by integrating over parameter uncertainty induced by finite data.

In **generative** settings (autoregressive text/code/audio/video) and **agentic** pipelines, test error alone is insufficient: behaviors are path-dependent and multi-modal. For sequences, our network induces factors $p(y_t \mid y_{<t}, x, W)$ and its sequence predictive distribution is $\int \prod_t p(y_t \mid y_{<t}, x, W)\, p(W \mid D)\, dW$, averaging the product over one shared parameter belief; token- or step-level uncertainties (e.g., predictive entropy per step) quantify diffusion of probability mass and guide temperature/top-p, tool calls, or conservative decoding. Formally, many evaluation targets reduce to expectations of task-specific functionals J over the posterior predictive of full outputs — e.g., $\mathrm{score}(x) = \mathbf{E}_{y_{1:T}\sim p(\bullet|x,D)}[J(y_{1:T})]$ — so a single Bayesian object $p(\bullet|x, D)$ supports diverse, multi-criterion assessments under one distribution. In agents, a policy $\pi_W(a \mid s)$ derived from the same DAG must

decide when to **abstain, route, or ask for help**; the Bayes action is the minimizer of the **posterior-predictive risk** $\mathbf{E}_{s\sim\text{env}}\mathbf{E}_{y\sim p(\cdot|s,D)}[\ell(a(s),y)]$. Thus, for both generative and agentic systems, the right object is not a point output but a **distribution** that composes across modules and time.

Finally, several operational pain points of non-Bayesian deep learning demands better solutions in the Bayesian framework and will motivate our development to integrate Bayesian inference into the standard deep learning modularized gradient update workflow leveraging auto-diff: **resource efficiency** (multi-sample inference incurs latency/memory; we will seek single-sweep posterior moment computation), **reliability under misspecification/shift** (posterior collapse, brittle priors, local Gaussianity failing under multi-modality-addressed by structured priors and geometry-aware updates), and **policy coupling & evaluation** (uncertainty only helps when it drives abstain/route/slow-down policies and is judged with decision-aware metrics; our construction produces interface-level uncertainties that plug directly into such policies).

We formalize the **modular posterior** over parameter blocks $W = \{W_b\}_{b\in\mathcal{B}}$ by an undirected dependency graph G : put an edge between b and b' if the log joint density contains a term that depends on W_b and $W_{b'}$ together (induced by the architecture, prior, or likelihood). The **Hammersley-Clifford theorem** applies: if $p(W \mid D)$ is strictly positive and **Markov with respect to** G — that is, for any disjoint node sets A, B, S, whenever S separates A and B in G we have the **conditional independence** $W_A \perp W_B \mid W_S$ — then the posterior **factorizes over the cliques** of G :

$$p(W \mid D) = \frac{1}{Z}\prod_{C\in\mathcal{C}} \phi_C\left(W_C\right), \quad \phi_C\left(W_C\right) > 0, \quad W_C = \{W_b : b \in C\},$$

For the exact inference statements, we additionally assume exponential-family potentials,

$$\phi_C\left(W_C\right) = \exp\left\{\langle\theta_C(D), T_C\left(W_C\right)\rangle\right\}$$

with sufficient statistics T_C and canonical parameters $\theta_C(D)$ that aggregate over the data (for a single datum (x, y) we write $\theta_C(x, y)$; over a dataset D, these contributions sum). After any required triangulation, we organize the cliques into a junction tree ($\mathcal{C}, \mathcal{S}$): a tree whose nodes are cliques and whose edges meet on separators $S = C \cap C'$. This preserves the product form and enables exact marginalization by message passing (variable elimination).

Junction-tree messages and beliefs. Let N_C be the separators adjacent to clique C, and N_S the cliques adjacent to separator S. The two message types are

$$\begin{aligned}
&\text{(clique } \rightarrow \text{ separator)} \quad m_{C\to S}\left(w_S\right) \propto \int \phi_C\left(w_C\right) \prod\nolimits_{S'\in N_C\setminus\{S\}} m_{S'\to C}\left(w_{S'}\right) dw_{C\setminus S},\\
&\text{(separator } \rightarrow \text{ clique)} \quad m_{S\to C}\left(w_S\right) \propto \prod\nolimits_{C'\in N_S\setminus\{C\}} m_{C'\to S}\left(w_S\right).
\end{aligned}$$

On a tree, one inward collect pass (leaves $\rightarrow$ root) followed by one outward distribute pass (root $\rightarrow$ leaves) using these equations yields exact clique/separator marginals as

$$b_C\left(w_C\right) \propto \phi_C\left(w_C\right) \prod_{S\in N_C} m_{S\to C}\left(w_S\right), \quad b_S\left(w_S\right) \propto \prod_{C\in N_S} m_{C\to S}\left(w_S\right)$$

We treat the parameters W as the random variables of interest; optional exogenous noises or latent activations can be added by augmenting G but are not required for what follows. We introduce dual coordinates (canonical and mean parameters) and express exact message passing as a forward moment computation followed by pseudo-gradient backpropagation. The backward corrections give the posterior moments in one update. A subsequent variational derivation relates the same posterior to the negative ELBO and a unit mirror step. The exact-tree result applies to any probabilistic model satisfying the representation assumptions below.

Recall **exponential-family, duality, regularity, Jacobians, and KL/Bregman**. For a regular (its natural-parameter set $\Theta = \{\eta : A(\eta) < \infty\}$ is open and A is C^2 on Θ) and minimal (the statistics are not affinely dependent, equivalently there is no nonzero vector u with $\langle u, T(z)\rangle$ almost surely constant) exponential family on a variable z with **sufficient statistics** $T(z)$, **canonical parameter** $\eta \in \mathbf{R}^d$, **log-partition** $A(\eta)$, and member

$$q_\eta(z) = \exp\{\langle\eta, T(z)\rangle - A(\eta)\}$$

Densities here use a fixed reference measure that absorbs the base measure. Work with nonredundant coordinates and mean parameters in $\nabla A(\Theta)$. The **mean (expectation) parameter** is $\mu = \mathbf{E}_{q_\eta}[T(z)] = \nabla_\eta A(\eta)$. The **convex conjugate (negative entropy)** $A^*(\mu) := \sup_\eta\{\langle \eta, \mu\rangle - A(\eta)\}$ gives the **mirror map** $\eta = \nabla_\mu A^*(\mu)$; the two maps are inverses. The Hessians/Jacobians are the **Fisher blocks**:

$$F(\eta) := \nabla^2_\eta A(\eta) = \frac{d\mu}{d\eta} \succ 0, \quad F(\mu)^{-1} := \nabla^2_\mu A^*(\mu) = \frac{d\eta}{d\mu} \succ 0. \tag{7}$$

Consequently, for any perturbations $d\eta, d\mu$ and covectors $\bar{\mu}, \bar{\eta}$,

$$\text{JVP} : d\mu = F(\eta)d\eta, \quad d\eta = F(\mu)^{-1}d\mu; \quad \text{VJP} : \bar{\eta} = F(\eta)\bar{\mu}, \quad \bar{\mu} = F(\mu)^{-1}\bar{\eta}$$

For any two members q_{η_1}, q_{η_2}, the **Bregman divergence** of A and the Bregman dual of A^* equals the KL:

$$\begin{aligned}\mathrm{KL}\left(q_{\eta_1} \| q_{\eta_2}\right) =& A\left(\eta_2\right) - A\left(\eta_1\right) - \left\langle \eta_2 - \eta_1, \nabla A\left(\eta_1\right)\right\rangle = D_A\left(\eta_2 \| \eta_1\right) \\ \mathrm{KL}\left(q_{\mu_1} \| q_{\mu_2}\right) =& A^*\left(\mu_1\right) - A^*\left(\mu_2\right) - \left\langle \nabla A^*\left(\mu_2\right), \mu_1 - \mu_2\right\rangle = D_{A^*}\left(\mu_1 \| \mu_2\right)\end{aligned}$$

We will instantiate these identities locally at each clique and separator region. The notation $F(\mu)$ abbreviates $F(\eta(\mu))$; redundant categorical coordinates require restriction to the simplex tangent.

Normalized exact-message computation. Choose a rooted junction-tree collect schedule and assign every model factor, including prior, evidence, and any separator potentials, to clique factors exactly once. At each clique, multiply its assigned factor by the incoming separator messages, normalize the product, and marginalize to the outgoing separator. Assume these products and marginals are represented exactly by regular minimal exponential families, with fixed compatible carriers and sufficient statistics $T_S = M_{C\to S}T_C$. The representations must remain exact in a neighborhood of each reached state. All cached messages are proper, with finite positive normalizers, interior mean parameters, and finite positive-definite Fisher blocks. The target clique and separator marginals are represented in these families and have finite sufficient-statistic moments. These conditions concern the chosen normalized schedule; global normalizability alone does not guarantee proper intermediate messages.

The computation stores each normalized message and its log scale separately, retaining every local log normalizer in the total log evidence $\ell = \log Z$. Let u index a directed clique or separator state, with forward-reference moments μ_u, Fisher matrix F_u, and exact posterior moments μ_u^{post}. Each state represents one tree cut: its remaining factors depend on its variables but are fixed when that normalized message is varied. In particular, a message is not duplicated as independent evidence. The forward references may contain partial evidence and need not agree with neighboring references. A proper prior can also be exposed as a normalized input, in which case its correction below is literally posterior minus prior moments.

For a moment-map edge $u \to v$, write $J_{v,u} = \partial\mu_v/\partial\mu_u$, and let $\mathrm{ch}(u)$ index the direct consumers of u's moments in this computation. Let g_u be the ordinary reverse-mode derivative of ℓ with respect to the saved μ_u, holding its associated cached log scale fixed; let g_u^{loc} collect direct derivatives from the local log normalizers and evidence terms. Define the moment pseudo-gradient $d_u = F_u g_u$. Thus g_u is a mean-coordinate derivative, while d_u is its Fisher conversion into a moment correction.

Theorem 3.1 (Exact message passing by pseudo-gradient backpropagation). *Under the normalized exact-message setup above, one reverse pass propagates*

$$d_u = F_u g_u^{\mathrm{loc}} + \sum_{v\in\mathrm{ch}(u)} F_u J_{v,u}^\top F_v^{-1} d_v = \mu_u^{\mathrm{post}} - \mu_u. \tag{8}$$

Consequently, the unit moment update

$$\boxed{\mu_u^{\mathrm{post}} = \mu_u + d_u} \tag{9}$$

gives the exact posterior clique and separator moments, which agree on every shared separator.

Proof. **1. A backward derivative is the posterior correction.** Cut the computation at a saved normalized state q_u. Exact elimination of the remaining factors gives a fixed outside message β_u and a cached scale s_u such that

$$Z = e^{s_u}\int q_u(w_u;\eta_u)\beta_u(w_u)\,dw_u, \qquad p_u^{\mathrm{post}}(w_u) = \frac{q_u(w_u;\eta_u)\beta_u(w_u)}{\int q_u(w_u;\eta_u)\beta_u(w_u)\,dw_u}.$$

Here s_u and β_u are held fixed in the derivative at this cut. The regular-family calculation remains finite in a neighborhood of the cut parameter, permitting differentiation under the integral and yielding

$$\nabla_{\eta_u} \ell = \mathbf{E}_{p_u^{\mathrm{post}}}[T_u] - \mathbf{E}_{q_u}[T_u] = \mu_u^{\mathrm{post}} - \mu_u = F_u \nabla_{\mu_u} \ell. \tag{10}$$

The last equality uses $\partial \mu_u / \partial \eta_u = F_u$. Because the mean-coordinate Fisher metric is F_u^{-1}, this is also the natural gradient of the log evidence in those coordinates. It is an exact finite moment difference.

2. Local product and marginal rules compute that derivative. At a product node C, let S_i index its incoming normalized separator messages, and put $M_i = M_{C \to S_i}$. Exact exponential-family multiplication gives

$$\eta_C = \vartheta_C + \sum_i M_i^\top \eta_{S_i}, \qquad \log z_C = A_C(\eta_C) - \sum_i A_{S_i}(\eta_{S_i}) + \mathrm{const}, \tag{11}$$

where ϑ_C is the fixed canonical contribution of the assigned local factor, and the constant accounts for fixed carrier factors. Therefore

$$J_{C,S_i} = F_C M_i^\top F_{S_i}^{-1}, \qquad \nabla_{\mu_{S_i}} \log z_C = F_{S_i}^{-1}(M_i \mu_C - \mu_{S_i}).$$

The derivative of the saved log normalizer and the adjoint through μ_C together give

$$\begin{aligned} g_{S_i} &= F_{S_i}^{-1}(M_i \mu_C - \mu_{S_i}) + F_{S_i}^{-1} M_i F_C g_C, \\ d_{S_i} &= (M_i \mu_C - \mu_{S_i}) + M_i d_C = M_i(\mu_C + d_C) - \mu_{S_i}. \end{aligned} \tag{12}$$

Thus a corrected clique moment gives the posterior correction at each incoming separator. The incoming q_{S_i} is generally not the marginal of this product q_C; the first term supplies precisely that difference. No identity $F_{S_i} = M_i F_C M_i^\top$ is used for these product inputs.

At a marginalization node, by contrast, q_S is the actual marginal of q_C. Hence $\mu_S = M_{C \to S} \mu_C$, and the reverse rule is

$$g_C = M_{C \to S}^\top g_S, \qquad d_C = F_C M_{C \to S}^\top F_S^{-1} d_S. \tag{13}$$

The outside factors depend on w_C only through w_S, so the normalizer at this cut depends on q_C through its exact marginal q_S. Its chain-rule derivative is therefore the clique correction in Eq. (10). Here the genuine marginal identity $F_S = M_{C \to S} F_C M_{C \to S}^\top$ also gives $M_{C \to S} d_C = d_S$.

3. One reverse pass and one finite update. At the root, the normalized clique product already contains all evidence, so its outside message is 1 and $g_{C_0} = d_{C_0} = 0$. Its saved log normalizer supplies the first nonzero derivatives to incoming separator messages. Equations (12)–(13), applied in reverse order through the collected computation, then give the exact correction at every state. More generally, ordinary reverse mode sums all direct and downstream contributions,

$$g_u = g_u^{\mathrm{loc}} + \sum_{v \in \mathrm{ch}(u)} J_{v,u}^\top g_v.$$

Multiplying by F_u gives Eq. (8). Each directed tree message is visited once in the collect pass and once in its reverse pass. Adding d_u once gives the posterior moments by Eq. (10); since they are marginals of the same posterior, $\mu_S^{\mathrm{post}} = M_{C \to S} \mu_C^{\mathrm{post}}$ on both sides of every separator. This proves the finite update and its consistency without assuming that the forward-reference messages were already calibrated. □

Variational interpretation: natural gradients and the unit mirror step

The preceding proof establishes the finite posterior correction computed by message backpropagation. We now give the complementary variational interpretation: the same posterior minimizes the junction-tree negative ELBO and is the endpoint of a unit mirror step. In the following variational derivation, μ denotes an arbitrary *consistent* reference collection in $\mathcal{M}_{\mathrm{JT}}$, rather than the partial-message references above. We write h for ordinary derivatives of Φ and $\delta\mu^\star$ for its constrained descent direction. These differ from the evidence adjoints g_u and the finite posterior corrections d_u in Theorem 3.1.

Setup and variational characterization. Let $(\mathcal{C}, \mathcal{S})$ be a junction tree over parameter blocks, with fixed exponential families $\{(T_C, A_C, A_C^*)\}_{C \in \mathcal{C}}$ for cliques and $\{(T_S, A_S, A_S^*)\}_{S \in \mathcal{S}}$ for separators.

Assume regular minimal families on fixed supports, with interior means and positive-definite Fisher blocks. For every clique–separator incidence, the sufficient statistics satisfy $T_S = M_{C\to S}T_C$, and the marginal of each admissible clique belief belongs to the stated separator family. Equality of separator means then implies equality of the separator distributions. Separators are indexed by tree edges, even when their variable sets repeat. For each region $r \in \mathcal{C} \cup \mathcal{S}$ let $\mu_r = \mathbf{E}[T_r]$ and $\eta_r = \nabla A_r^*(\mu_r)$. The **consistent mean domain** is the set of locally consistent means

$$\mathcal{M}_{\mathrm{JT}} = \{\mu = \{\mu_C, \mu_S\} : \mu_C \in \Delta_C^\circ, \mu_S \in \Delta_S^\circ, \mu_S = M_{C\to S}\mu_C \forall (C,S)\}, \tag{14}$$

where $M_{C\to S}$ is the fixed linear marginalization map taking clique sufficient-statistic means to the corresponding separator means. Here $\Delta_r^\circ = \nabla A_r(\Theta_r)$ is the admissible interior mean domain. Full positive categorical and nonsingular full Gaussian families satisfy the stated compatibility assumptions in their usual moment coordinates. Assemble the model-induced canonical parameters $\theta = \{\theta_C, \theta_S\}$ (aggregated over the dataset D). With respect to compatible product dominating measures, write $q_r = \kappa_r \exp\{\langle \eta_r, T_r\rangle - A_r(\eta_r)\}$ with fixed carrier functions κ_r. Require $p_\theta \propto (\prod_C \kappa_C / \prod_S \kappa_S)\exp\{\sum_r \langle \theta_r, T_r\rangle\}$ to be normalizable and to have represented clique and separator marginals. Unit carriers give the density convention above; the clique-potential representation has $\theta_S = 0$. The **negative ELBO** reads

$$\Phi(\mu) = -\sum_{r\in\mathcal{C}\cup\mathcal{S}} \langle \theta_r, \mu_r\rangle + \sum_{C\in\mathcal{C}} A_C^*(\mu_C) - \sum_{S\in\mathcal{S}} A_S^*(\mu_S), \qquad \mu \in \mathcal{M}_{\mathrm{JT}}. \tag{15}$$

Under these exact-family assumptions, $\Phi(\mu)$ equals $\mathrm{KL}(q_\mu \| p_\theta)$ up to an additive constant, where q_μ is the globally consistent distribution reconstructed from region beliefs; its unique minimizer corresponds to the represented exact posterior region means. This conclusion follows from the KL identity and does not require a global convexity assertion for a restricted mean domain.

Mean-parameter propagation with normalized conditional interfaces. Fixing an orientation of the junction tree with an arbitrary root clique C_0, fix an interior consistent reference μ. For a non-root clique C with parent separator S_p and child separators $\{S_c\}$:

- Forward (root $\to$ leaves) mean propagation. Holding the reference conditional q_C/q_{S_p} fixed while changing the incoming separator belief gives the normalized extension

$$q'_C(w_C) = q_C(w_C)\frac{q'_{S_p}(w_{S_p})}{q_{S_p}(w_{S_p})}. \tag{16}$$

  In canonical and mean coordinates,

$$\begin{aligned} \eta'_{S_p} &= \nabla A^*_{S_p}(\mu'_{S_p}), \quad \eta'_C = \eta_C + M^\top_{C\to S_p}(\eta'_{S_p} - \eta_{S_p}), \\ \mu'_C &= \nabla A_C(\eta'_C), \quad \mu'_{S_c} = M_{C\to S_c}\mu'_C \quad \text{for each child separator } S_c. \end{aligned} \tag{17}$$

  Starting from an admissible root belief and applying these extensions outward preserves consistency whenever all reached beliefs remain in the declared domain. The fixed reference conditional parameter $\eta_C - M^\top_{C\to S_p}\eta_{S_p}$ is distinct from the target potential parameter θ_C. At the reference,

$$L_C := \frac{\partial \mu'_C}{\partial \mu'_{S_p}} = F_C M^\top_{C\to S_p} F_{S_p}^{-1}, \qquad F_{S_p} = M_{C\to S_p} F_C M^\top_{C\to S_p}. \tag{18}$$

- Backward (leaves $\to$ root) adjoint propagation. Reverse-mode differentiation of these moment maps gives the recursive accumulation in Eq. (19).

In both directions, the fixed linear maps are $M_{C\to S}$ and their adjoints $M^\top_{C\to S}$: $M_{C\to S}$ performs **marginalization of sufficient-statistic means** (pushes means from a clique down to its interface), while $M^\top_{C\to S}$ is the **adjoint** that injects separator canonical parameters into the clique's canonical space. These are **constant Jacobians** that encode graph topology; all **curvature** in the computation comes from the **mirror maps** $\eta = \nabla A^*(\mu)$ and $\mu = \nabla A(\eta)$. This separation gives the variational adjoints and constrained natural-gradient direction below.

The local derivatives of Φ are

$$h_C^{\mathrm{loc}} = \eta_C - \theta_C, \qquad h_S^{\mathrm{loc}} = -\eta_S - \theta_S.$$

Reverse-mode differentiation of the normalized conditional moment graph gives

$$\begin{aligned} h_C^{\mathrm{tot}} &= h_C^{\mathrm{loc}} + \sum_{S_c \text{ child of } C} M_{C\to S_c}^{\top} h_{S_c}^{\mathrm{tot}}, \\ h_{S_p}^{\mathrm{tot}} &= h_{S_p}^{\mathrm{loc}} + F_{S_p}^{-1} M_{C\to S_p} F_C h_C^{\mathrm{tot}}. \end{aligned} \tag{19}$$

For each nonroot clique define its conditional Fisher block

$$K_C = F_C - F_C M_{C\to S_p}^{\top} F_{S_p}^{-1} M_{C\to S_p} F_C. \tag{20}$$

The constrained natural-gradient direction is reconstructed from root to leaves:

$$\begin{aligned} \delta\mu_{C_0}^{\star} &= -F_{C_0} h_{C_0}^{\mathrm{tot}}, \\ \delta\mu_{S_p}^{\star} &= M_{P\to S_p} \delta\mu_P^{\star}, \\ \delta\mu_C^{\star} &= L_C \delta\mu_{S_p}^{\star} - K_C h_C^{\mathrm{tot}} \quad (C \neq C_0), \end{aligned} \tag{21}$$

where P is the parent clique of C. As proved below, this is the unique solution to

$$\begin{aligned} &\min_{\delta\mu} \langle \nabla\Phi(\mu), \delta\mu \rangle + \tfrac{1}{2} \delta\mu^{\top} H(\mu) \delta\mu \quad \text{s.t. } B\delta\mu = 0, \\ &H = \mathrm{diag}\big(\{F_C^{-1}\}_C, \{-F_S^{-1}\}_S\big). \end{aligned} \tag{22}$$

Here B has a row $[-M_{C\to S} \mid I]$ for every clique–separator incidence. The signed matrix H is positive definite on the consistency tangent. This direction is the local tangent of a mirror path; the finite unit mirror minimizer is the same posterior computed by Theorem 3.1.

Proof of the variational characterization. We use two elementary ingredients: (i) the compatibility of clique/separator exponential families with the linear marginal maps (so that marginal means are linear images of clique means), and (ii) the KKT conditions for the constrained steepest-descent direction in the Fisher metric.

1. Feasibility of the forward map.

The reference q_{S_p} is the actual marginal of q_C, hence

$$\int q_C(w_C) \frac{q'_{S_p}(w_{S_p})}{q_{S_p}(w_{S_p})} \, dw_{C\setminus S_p} = q'_{S_p}(w_{S_p}).$$

Thus Eq. (16) is normalized and has the requested incoming marginal. With $\Delta\eta_{S_p} = \eta'_{S_p} - \eta_{S_p}$, statistic compatibility gives

$$A_C(\eta_C + M_{C\to S_p}^{\top} \Delta\eta_{S_p}) - A_C(\eta_C) = A_{S_p}(\eta'_{S_p}) - A_{S_p}(\eta_{S_p}),$$

which proves the canonical formula in Eq. (17). The EF Jacobians give L_C in Eq. (18); taking covariances of $T_{S_p} = M_{C\to S_p} T_C$ under the consistent reference gives the stated Fisher identity. In particular,

$$M_{C\to S_p} L_C = I, \qquad M_{C\to S_p} K_C = 0.$$

Applying the normalized extensions down the tree proves feasibility. Their transpose chain rule gives Eq. (19).

2. Backward adjoints and the projected natural gradient. Let $h = \nabla\Phi(\mu)$ and $H = \mathrm{diag}\big(\{F_C^{-1}\}_C, \{-F_S^{-1}\}_S\big)$.

Every feasible direction has the decomposition $\delta\mu_C = L_C \delta\mu_{S_p} + e_C$ with $M_{C\to S_p} e_C = 0$. Because $L_C^{\top} F_C^{-1} = F_{S_p}^{-1} M_{C\to S_p}$, the cross term vanishes and

$$\delta\mu^{\top} H \, \delta\mu = \delta\mu_{C_0}^{\top} F_{C_0}^{-1} \delta\mu_{C_0} + \sum_{C\neq C_0} e_C^{\top} F_C^{-1} e_C. \tag{23}$$

Each nonroot clique contributes one parent-separator quadratic, which cancels the corresponding negative separator term, including at branches. The expression is strictly positive for every nonzero feasible direction: a zero root change and zero e_C's imply zero changes everywhere by outward reconstruction.

The KKT system reads

$$H\delta\mu + h + B^\top\lambda = 0, \qquad B\delta\mu = 0. \tag{24}$$

For each edge S joining parent clique P and child clique C, assign two incidence multipliers

$$\lambda_{PS} = -h_S^{\mathrm{tot}}, \qquad \lambda_{CS} = F_S^{-1}\delta\mu_S^\star + L_C^\top h_C^{\mathrm{tot}}.$$

Substitution of Eq. (21) and the backward recurrence gives the clique and separator KKT equations

$$\begin{aligned} F_C^{-1}\delta\mu_C^\star + h_C^{\mathrm{loc}} - \sum_{S\in N_C} M_{C\to S}^\top\lambda_{CS} = 0, \\ -F_S^{-1}\delta\mu_S^\star + h_S^{\mathrm{loc}} + \lambda_{PS} + \lambda_{CS} = 0. \end{aligned} \tag{25}$$

Indeed, $F_C^{-1}\delta\mu_C^\star = -h_C^{\mathrm{tot}} + M_{C\to S_p}^\top\lambda_{CS_p}$ for a nonroot clique, while the root has no parent term; the child-edge multipliers supply the remaining terms. At a separator, $h_S^{\mathrm{tot}} = h_S^{\mathrm{loc}} + L_C^\top h_C^{\mathrm{tot}}$ gives the second equation. The identities $ML_C = I$ and $MK_C = 0$ give $B\,\delta\mu^\star = 0$. Positive definiteness on this tangent then proves that the reconstructed direction is the unique constrained minimizer.

3. Unit mirror step gives exact posterior means.

The unit entropy-mirror update is

$$\begin{aligned} \mu^+ &= \arg\min_{\nu\in\mathcal{M}_{\mathrm{JT}}} \langle\nabla\Phi(\mu), \nu-\mu\rangle + D_{A_{\mathsf{T}}^*}(\nu\|\mu), \\ A_{\mathsf{T}}^* &= \sum_C A_C^* - \sum_S A_S^*. \end{aligned} \tag{26}$$

Tree reconstruction gives $q_\mu = \prod_C q_{\mu,C} / \prod_S q_{\mu,S}$. With exact region marginals and the fixed carriers,

$$\begin{aligned} D_{A_{\mathsf{T}}^*}(\nu\|\mu) &= \sum_C \mathrm{KL}(q_{\nu,C}\|q_{\mu,C}) - \sum_S \mathrm{KL}(q_{\nu,S}\|q_{\mu,S}) \\ &= \mathrm{KL}(q_\nu\|q_\mu). \end{aligned}$$

Thus H is the joint Fisher metric on consistent moment changes. Since $\Phi(\mu) = A_{\mathsf{T}}^*(\mu) - \langle\theta,\mu\rangle$, the unit mirror objective satisfies

$$\langle\nabla\Phi(\mu), \nu-\mu\rangle + D_{A_{\mathsf{T}}^*}(\nu\|\mu) = \Phi(\nu) - \Phi(\mu).$$

The assumed target representation gives $\Phi(\nu) = \mathrm{KL}(q_\nu\|p_\theta) + \mathrm{const}$. Its minimum is attained uniquely at the represented target distribution and hence at its region means. This proves the finite endpoint without identifying its multipliers with those of the tangent quadratic.

For the local tangent, let μ_ρ be the minimizer near μ of

$$\rho\langle h, \nu-\mu\rangle + D_{A_{\mathsf{T}}^*}(\nu\|\mu), \qquad \nu\in\mathcal{M}_{\mathrm{JT}}, \qquad \mu_0 = \mu. \tag{27}$$

Each marginal map has full row rank by the Fisher identity; eliminating leaf incidences therefore shows that B has full row rank. Together with Eq. (23), this makes the KKT matrix nonsingular, so the implicit-function theorem supplies a unique smooth local branch. Differentiating its stationarity equations at $\rho = 0$ gives Eq. (24) for $d\mu_\rho/d\rho|_{\rho=0}$, which is therefore $\delta\mu^\star$. No claim that every intermediate finite mirror solution remains in a restricted family is needed.

□

The discrete HMM and linear–Gaussian chain examples are immediate specializations of Theorem 3.1 to the case where the junction tree is a simple chain and the local exponential families are exact. Collect from C_1 toward root C_T, with S_t joining C_t to C_{t+1}. Each q_{S_t} is the outgoing marginal of the collected q_{C_t}. The backward correction starts with $d_{C_T} = 0$ and, for $t = T-1,\dots,1$, uses

$$\begin{aligned} d_{S_t} &= M_{C_{t+1}\to S_t}(\mu_{C_{t+1}} + d_{C_{t+1}}) - \mu_{S_t}, \\ d_{C_t} &= F_{C_t} M_{C_t\to S_t}^\top F_{S_t}^{-1} d_{S_t}. \end{aligned}$$

The first line includes the next product's log-normalizer derivative; the second differentiates the actual outgoing marginal. Thus $\mu_C + d_C$ and $\mu_S + d_S$ are the exact posterior moments after this reverse pass.

Discrete HMM / forward-backward. *For a chain with cliques $C_t = \{z_{t-1}, z_t\}$ and separators $S_t = \{z_t\}$ where $z_t \in \{1, \dots, K\}$ and with the exact categorical families on pairs and nodes, one reverse pass followed by the unit moment update gives the exact posterior marginals (γ_t, ξ_t) computed by the classical forward–backward algorithm.*

Proof. Use the chain recurrence above, with the prior included in the first collected clique. Using the HMM maps $M_{C_t \to S_t}$ as column sums and $M_{C_t \to S_{t-1}}$ as row sums (and their Euclidean adjoints as column/row broadcasts), and the categorical Fisher blocks F_{C_t} and $F_{S_t}^{-1}$ in minimal coordinates, Theorem 3.1 gives $\mu_{C_t}^{\mathrm{post}} = \mu_{C_t} + d_{C_t}$ and $\mu_{S_t}^{\mathrm{post}} = \mu_{S_t} + d_{S_t}$. These are the HMM posterior pair and node marginals (ξ_t, γ_t). Row/column sums and categorical Fisher actions can be applied without forming dense Fisher matrices, yielding the classical $O\left(TK^2\right)$ time and $O\left(TK^2\right)$ storage. □

Linear-Gaussian chain / Kalman–RTS. *For a nonsingular linear-Gaussian state-space chain with cliques $C_t = \{x_{t-1}, x_t\}$, separators $S_t = \{x_t\}$, and full Gaussian families for cliques/nodes, one reverse pass followed by the unit moment update gives the exact smoothed first and second moments, including adjacent cross moments, computed by Kalman filtering and Rauch–Tung–Striebel smoothing.*

Proof. Using Gaussian mean parameters $\mu_{S_t} = (m_t, M_t)$ with $M_t = \mathbf{E}[x_t x_t^\top]$ and μ_{C_t} including $M_{t-1,t} = \mathbf{E}[x_{t-1} x_t^\top]$, use the same chain recurrence with proper filtering messages, including the initial prior. Here $M_{C_t \to S_{t-1}}$ and $M_{C_t \to S_t}$ pick out the corresponding first/second-moment blocks, while the Gaussian Fisher blocks $F_{S_t}^{-1}$ and F_{C_t} are the information-form Jacobians for node and pair moments (as fixed earlier). Theorem 3.1 gives the exact smoothed moment collection by $\mu_u^{\mathrm{post}} = \mu_u + d_u$, hence the same moments as Kalman–RTS. Applying the Gaussian Fisher actions through means and covariances yields $O\left(Td^3\right)$ time and $O\left(Td^2\right)$ storage. □

B Projected EP and Bayesified Moment Graphs

This section gives the formal version of Theorem 3.2 and the module-level consequences used by Algorithm 1. The main idea is that EP moment matching is a unit natural-gradient step in mean-parameter space. A Bayesified module transports the corresponding signal: it receives a child correction, differentiates its forward moment map, and returns corrections for its parents and local parameter beliefs. The exact tilted-moment interpretation requires the computation to evaluate the stated fixed-message normalizer.

B.1 Projected EP as a natural-gradient moment update

Let u be an uncertain block represented by a regular minimal exponential-family belief

$$q_u(z;\eta_u) = h_u(z)\exp\{\langle \eta_u, T_u(z)\rangle - A_u(\eta_u)\}, \tag{28}$$

with selected sufficient statistics T_u, canonical parameter η_u, mean parameter $\mu_u = \mathbb{E}_{q_u}[T_u(z)]$, and Fisher block $F_u = \partial\mu_u/\partial\eta_u = \mathrm{Cov}_{q_u}[T_u(z)]$. Work at an interior belief, so F_u is invertible. Let $\beta_u(z) \geq 0$ be a downstream likelihood message held fixed with respect to η_u during the local update, with the other beliefs also fixed. Define the local tilted normalizer and tilted distribution by

$$Z_u(\eta_u) = \int \beta_u(z) q_u(z;\eta_u)\, dz, \qquad \widehat{p}_u(z) = Z_u(\eta_u)^{-1}\beta_u(z)q_u(z;\eta_u). \tag{29}$$

Assume $0 < Z_u < \infty$, the selected tilted moments are finite, and differentiation under the integral is valid. Projected EP updates q_u by matching the selected tilted moments, $\mu_u^\star = \mathbb{E}_{\widehat{p}_u}[T_u(z)]$. The projected belief exists when these moments lie in the chosen family's mean-parameter domain.

Theorem 3.2 (Projected EP; formal restatement)**.** *For the block u above,*

$$\mu_u^\star = \mu_u + F_u \nabla_{\mu_u} \log Z_u. \tag{30}$$

Equivalently, if $g_u = \nabla_{\mu_u} \log Z_u$, then the EP moment increment is $\Delta\mu_u = \mu_u^\star - \mu_u = F_u g_u$. Thus projected EP moment matching is a unit natural-gradient step in mean-parameter space, where the Fisher metric in mean coordinates is F_u^{-1}.

Here g_u is an ordinary mean-coordinate derivative; the reference belief q_u determines both F_u and the starting moments μ_u. The correction $\Delta\mu_u$ is the finite change to its local tilted projection. Let a differentiable directed acyclic moment computation evaluate a scalar ℓ, including all direct dependence on its belief states. The decoded belief states have interior coordinates and invertible Fisher blocks. At the cached forward state, write $J_{c,u} = \partial\mu_c/\partial\mu_u$ and let g_u^{direct} collect derivatives of objective terms attached directly to u. Reverse accumulation, with every downstream use included, gives

$$g_u = g_u^{\mathrm{direct}} + \sum_{c\in\mathrm{ch}(u)} J_{c,u}^\top g_c, \tag{31}$$

$$d_u = F_u g_u^{\mathrm{direct}} + \sum_{c\in\mathrm{ch}(u)} F_u J_{c,u}^\top F_c^{-1} d_c, \qquad d_u = F_u g_u. \tag{32}$$

For a block u, if the composed scalar equals its fixed-message $\log Z_u$, up to a term independent of μ_u under the stated update schedule, then $d_u = \mu_u^\star - \mu_u$. A declared approximate normalizer gives the corresponding Fisher-scaled derivative; exact forward moments alone do not establish equality with the original tilted moments.

Proof. Differentiating the tilted normalizer with respect to the canonical parameter gives

$$\nabla_{\eta_u} \log Z_u = \frac{1}{Z_u}\int \beta_u(z) q_u(z;\eta_u) \nabla_{\eta_u} \log q_u(z;\eta_u)\, dz. \tag{33}$$

For the exponential family in Eq. (28), $\nabla_{\eta_u} \log q_u(z;\eta_u) = T_u(z) - \nabla_{\eta_u} A_u(\eta_u) = T_u(z) - \mu_u$. Hence

$$\nabla_{\eta_u} \log Z_u = \mathbb{E}_{\widehat{p}_u}[T_u(z)] - \mu_u = \mu_u^\star - \mu_u. \tag{34}$$

The chain rule gives $\nabla_{\eta_u} \log Z_u = (\partial\mu_u/\partial\eta_u)\nabla_{\mu_u} \log Z_u = F_u \nabla_{\mu_u} \log Z_u$. Combining this identity with Eq. (34) proves Eq. (30). A damped update $\mu_u^+ = (1-\rho)\mu_u + \rho\mu_u^\star$ is valid when this

segment lies in the mean-parameter domain. For the Gaussian families used here, valid endpoints preserve covariance validity for $0 \le \rho \le 1$; see Eq. (54). Reverse induction through the computation graph gives Eq. (31), including the sum over shared uses. Substituting $g_c = F_c^{-1} d_c$ and multiplying by F_u gives Eq. (32). The score identity then identifies this signal with the tilted-moment correction at every block satisfying the stated normalizer condition. Applying the same chain rule to a declared approximate normalizer gives its approximate correction. □

Explicit-site EP bookkeeping. Suppose $q(u) \propto p_0(u) \prod_s \widetilde{f}_s(u)$, with $\eta_q = \eta_0 + \sum_s \lambda_s$. For a factor f_s and power $\alpha > 0$, form the cavity $\eta^- = \eta_q - \alpha\lambda_s$ and the tilt $\widehat{p}(u) \propto q^-(u) f_s(u)^\alpha$. The theorem applies with $q_u = q^-$ and fixed message $\beta_u = f_s^\alpha$. If its projected tilted belief has canonical parameter $\eta^\star$, the site target is

$$\lambda_s^{\mathrm{new}} = \alpha^{-1}(\eta^\star - \eta^-), \tag{35}$$

With site damping, $\lambda_s^+ = (1-\rho)\lambda_s + \rho\lambda_s^{\mathrm{new}}$. Reconstructing the global belief under the stated product convention gives

$$\eta_q^+ = \eta_q + \rho(\lambda_s^{\mathrm{new}} - \lambda_s) = \eta_q + \frac{\rho}{\alpha}(\eta^\star - \eta_q). \tag{36}$$

Thus $\eta_q^+ = \eta^\star$ for an undamped ordinary EP update ($\alpha = \rho = 1$). For other powers, the tilted projection and the reconstructed global belief need not coincide. The cavity, projection, and reconstructed belief must remain valid. Canonical site damping also differs from raw-moment damping in Eq. (5).

Corollary (tied power-EP with a fixed prior). Let $q = q_\eta$ and $p_0 = q_{\eta_0}$ be normalized beliefs in the same regular minimal exponential family, with fixed carrier and support. Write $\mu = \nabla A(\eta)$ and $F = \nabla^2 A(\eta)$. For fixed $\tau > 0$, N likelihood factors $f_n(w) > 0$, and $0 < \alpha < N$, set $k = \alpha/N$ and

$$\eta_c = (1-k)\eta + k\eta_0, \qquad q_c = \frac{q^{1-k} p_0^k}{C_\alpha}, \qquad C_\alpha = \int q^{1-k} p_0^k.$$

In tied-site coordinates $\eta = \eta_0 + N\lambda$, this gives $\eta_c = \eta_0 + (N-\alpha)\lambda$, retaining the prior once. Define $Z_n^c = \mathbb{E}_{q_c}[f_n^{\alpha/\tau}]$ and $\mu_n^\star = \mathbb{E}_{q_c f_n^{\alpha/\tau}/Z_n^c}[T]$. A tied α-divergence energy [11] is

$$\mathcal{E}_\alpha(\mu) = -\frac{\tau}{\alpha}\left[\frac{1}{N}\sum_{n=1}^N \log Z_n^c + \log C_\alpha\right].$$

Assume positive finite normalizers, finite tilted moments, and valid differentiation under the integrals. Let $g_{\mathrm{tied}} = -\nabla_\mu \mathcal{E}_\alpha$ be the negative gradient in mean coordinates. Then

$$d_{\mathrm{tied}} = F g_{\mathrm{tied}} = \frac{\tau(1-k)}{\alpha}\left(\frac{1}{N}\sum_{n=1}^N \mu_n^\star - \mu\right). \tag{37}$$

Proof. With $\mu_c = \nabla A(\eta_c)$, the score identity gives

$$\nabla_\eta \log Z_n^c = (1-k)(\mu_n^\star - \mu_c), \qquad \nabla_\eta \log C_\alpha = (1-k)(\mu_c - \mu).$$

The second identity also follows from $\log C_\alpha = A(\eta_c) - (1-k)A(\eta) - kA(\eta_0)$. Adding the derivatives cancels μ_c; using $\nabla_\eta = F\nabla_\mu$ proves the result. □

Uniform minibatches replace the dataset mean of $\log Z_n^c$ by the batch mean and retain $\log C_\alpha$ once, with N fixed. The equality above uses exact normalizers; approximate normalizers give derivatives of their declared approximation. The finite belief update must be specified separately and remain in the belief domain. For fixed τ, if differentiation at $\alpha = 0$ is also valid,

$$\lim_{\alpha \downarrow 0} \mathcal{E}_\alpha = -\frac{1}{N}\sum_{n=1}^N \mathbb{E}_q[\log f_n] + \frac{\tau}{N}\mathrm{KL}(q\|p_0).$$

Indeed, the derivatives at zero of $\log C_\alpha$ and $\log Z_n^c$ are $-\mathrm{KL}(q\|p_0)/N$ and $\mathbb{E}_q[\log f_n]/\tau$.

B.2 Diagonal Gaussian messages

For $q(u) = \mathcal{N}(m, \operatorname{Diag} v)$, the stored coordinates are (m, v), while the Gaussian mean parameters are (m, s) with $s = m^{\odot 2} + v$. Theorem 3.2 gives the conversion used in the main algorithm.

Corollary B.1 (Diagonal Gaussian moment increment)**.** *Let $g_m = \partial \ell / \partial m$ and $g_v = \partial \ell / \partial v$, where $\ell = \log Z_u$ and the derivatives are with respect to the stored coordinates (m, v). The EP moment increment in (m, s) coordinates is*

$$\Delta m = v \odot g_m, \qquad \Delta s = 2m \odot v \odot g_m + 2v^{\odot 2} \odot g_v. \tag{38}$$

Conversely, if a module receives a moment increment $(\Delta m, \Delta s)$, the adjoints with respect to the stored coordinates are

$$g_m = \Delta m \oslash v, \qquad g_v = (\Delta s - 2m \odot \Delta m) \oslash (2v^{\odot 2}). \tag{39}$$

The two conversions are inverses whenever v is strictly positive.

Proof. Apply Theorem 3.2 to the scalar sufficient statistics $T(u) = (u, u^2)$ coordinatewise. The canonical Fisher block $\operatorname{Cov}[T(u)]$, expressed using m, v, is

$$F_{(m,s)} = \begin{pmatrix} v & 2mv \\ 2mv & 4m^2 v + 2v^2 \end{pmatrix}. \tag{40}$$

Its inverse is the Fisher metric in mean coordinates (m, s). The storage map is $(m, v) \mapsto (m, s = m^2 + v)$. Therefore $\partial \ell / \partial s = \partial \ell / \partial v = g_v$ and $\partial \ell / \partial m$ in mean coordinates is $g_m - 2m g_v$. Multiplying by Eq. (40) gives $\Delta m = v g_m$ and $\Delta s = 2mv g_m + 2v^2 g_v$ coordinatewise, which proves Eq. (38). Solving these equations for (g_m, g_v) gives Eq. (39). □

Coordinate convention. The Fisher map acts on raw mean coordinates (m, s), whereas a practical module differentiates its stored forward map in (m, v). Decode its incoming $(\Delta m, \Delta s)$ by Eq. (39), take the vector–Jacobian product in stored coordinates, and encode each result by Eq. (38). In particular, Δs is not a variance increment; the finite variance update follows from $v^+ = s^+ - (m^+)^{\odot 2}$ as in Eq. (5). Activation states transport corrections; persistent parameter beliefs are updated once after contributions from their shared uses are summed.

B.3 Module-level Bayesified backpropagation

A Bayesified module has a forward moment map $\mu_c = \mathcal{M}_r(\{\mu_p\}_{p \in \mathrm{pa}(r)}, \{\mu_w\}_{w \in \mathrm{par}(r)})$ from parent and local parameter moments to child moments. The module receives a child moment increment $\Delta \mu_c$, converts it into a child adjoint, differentiates its own moment map, and then converts the resulting adjoints into moment increments for its parents and parameter beliefs.

Proposition B.2 (Module-level moment backpropagation)**.** *Let F_c be the Fisher block for the child belief. For a downstream scalar calculation ℓ, suppose the incoming signal at the cached forward state is $\Delta \mu_c = F_c \nabla_{\mu_c} \ell$. Define the child adjoint $g_c = F_c^{-1} \Delta \mu_c$. Let $J_{r,p} = \partial \mu_c / \partial \mu_p$ and $J_{r,w} = \partial \mu_c / \partial \mu_w$. The module returns*

$$g_p = J_{r,p}^\top g_c, \qquad g_w = J_{r,w}^\top g_c, \qquad \Delta \mu_p = F_p g_p, \qquad \Delta \mu_w = F_w g_w. \tag{41}$$

This is the natural-gradient adjoint of the projected moment map. For a parent or parameter block u, it equals the EP tilted-moment correction when the composed scalar calculation is the fixed-message $\log Z_u$ of Eq. (29), up to a term independent of μ_u. An approximate calculation instead gives the Fisher-scaled derivative of its declared surrogate. Exact forward moments alone do not guarantee that the resulting normalizer is exact.

Proof. The incoming moment increment satisfies $\Delta \mu_c = F_c \nabla_{\mu_c} \ell$, so $g_c = \nabla_{\mu_c} \ell$. The chain rule through the forward moment map gives $\nabla_{\mu_p} \ell = J_{r,p}^\top g_c$ and $\nabla_{\mu_w} \ell = J_{r,w}^\top g_c$. Multiplication by F_p and F_w gives Eq. (41). When the stated normalizer condition holds, Theorem 3.2 identifies these signals with tilted-moment differences. Otherwise the chain-rule calculation remains valid for the chosen scalar, without that additional probabilistic identification. □

Corollary B.3 (Affine module)**.** *For $z = Wa + b$ with independent diagonal Gaussian beliefs $q(a) = \mathcal{N}(m_a, \operatorname{Diag} v_a)$, $q(W) = \mathcal{N}(M, \operatorname{Diag} V_W)$, and $q(b) = \mathcal{N}(m_b, \operatorname{Diag} v_b)$, the selected first and diagonal second moments are*

$$m_z = Mm_a + m_b, \qquad v_z = M^{\odot 2} v_a + V_W(m_a^{\odot 2} + v_a) + v_b. \tag{42}$$

The backward rule of Proposition B.2 applied to Eq. (42) *returns an upstream moment increment for a and local moment increments for W and b when b is Bayesian. Convolution, embedding, and adapter projections use the same result with the corresponding linear operator in place of matrix multiplication.*

Proof. The mean formula follows from independence: $\mathbb{E}[z] = \mathbb{E}[W]\mathbb{E}[a] + \mathbb{E}[b] = Mm_a + m_b$. For each output coordinate, $\operatorname{Var}(z_i) = \operatorname{Var}(\sum_j W_{ij}a_j + b_i)$. Under the independent diagonal approximation, $\operatorname{Var}(W_{ij}a_j) = M_{ij}^2 v_{a,j} + V_{W,ij}\mathbb{E}[a_j^2]$, and $\mathbb{E}[a_j^2] = m_{a,j}^2 + v_{a,j}$. Summing over j and adding $v_{b,i}$ gives Eq. (42). This is exact for the retained first and diagonal second moments under the stated independent Gaussian beliefs; it is not a full joint-Gaussian claim about the bilinear product Wa. □

For an explicit affine backward rule, decode the incoming correction into $r = \partial\ell/\partial m_z$ and $t = \partial\ell/\partial v_z$. Differentiating Eq. (42) gives

$$\begin{aligned} g_{m_a} &= M^\top r + 2m_a \odot (V_W^\top t), & g_{v_a} &= (M^{\odot 2} + V_W)^\top t, \\ g_M &= r m_a^\top + 2M \odot (t v_a^\top), & g_{V_W} &= t(m_a^{\odot 2} + v_a)^\top, \\ g_{m_b} &= r, & g_{v_b} &= t. \end{aligned} \tag{43}$$

Each pair is converted to a raw-moment correction by Eq. (38). These expressions follow by direct differentiation of the mean and variance terms; shared convolutional or embedding parameters accumulate contributions over all their uses.

Gaussianized observation head. Cache an expansion point $\bar{m}$, $\bar{p} = \operatorname{softmax}(\bar{m})$, $\bar{J} = \operatorname{Diag}(\bar{p}) - \bar{p}\bar{p}^\top$, and $\bar{c} = \bar{p} - \bar{J}\bar{m}$. Bars denote quantities held fixed during the reverse calculation. The local observation model has message $\beta(z) = \mathcal{N}(y; \bar{c} + \bar{J}z, RI)$ with $R > 0$. For $q(z) = \mathcal{N}(m, V)$, its exact normalizer is

$$Z(m, V) = \mathcal{N}(y; \bar{c} + \bar{J}m, S), \qquad S = \bar{J}V\bar{J}^\top + RI. \tag{44}$$

Writing $e = y - \bar{c} - \bar{J}m$ and $a = S^{-1}e$, differentiation gives

$$\nabla_m \log Z = \bar{J}^\top a, \qquad \nabla_V \log Z = \tfrac{1}{2}\bar{J}^\top(aa^\top - S^{-1})\bar{J}. \tag{45}$$

Gaussian Fisher conversion gives the same first- and second-moment corrections as conditioning with $K = V\bar{J}^\top S^{-1}$, $m^\star = m + Ke$, and $V^\star = V - K\bar{J}V$; the full-covariance conversion is derived in Appendix B.5. At $m = \bar{m}$ this is the head in the main text. A diagonal output belief retains the diagonal second moments of this tilted Gaussian. The identity is exact for this fixed local observation model; it does not replace the original softmax likelihood by an exact Gaussian likelihood. Differentiating the cached expansion itself would define a different scalar calculation.

Other differentiable modules use the same decode–differentiate–encode rule with their declared analytic or approximate moments. This includes nonlinear activations and successive adapter factors; it does not require a separate message derivation for every layer. Any cached linearization is held fixed or differentiated according to the stated moment map. Contributions at a reused activation or parameter state are summed before its local Fisher conversion.

B.4 Shared variables in residual and concatenation primitives

Residual and concatenation modules reuse an activation inside a deterministic graph fragment. The clean probabilistic representation keeps that activation as one variable. This subsection formalizes why the module rule then counts the cavity once and why attaching a cavity density to every equalized copy changes the tilted distribution.

Proposition B.4 (Shared-variable primitives count the cavity once)**.** *Let x have cavity belief $q_x(x;\eta_x)$ and let w have optional cavity belief $q_w(w;\eta_w)$. The cavity blocks are independent. Let a deterministic graph fragment produce $y = G(x,w)$, where G may use x multiple times internally. Let $\beta_y(y)$ be a fixed downstream likelihood message, and assume the normalizer and derivatives satisfy the conditions of Theorem 3.2. The shared-variable tilted normalizer is*

$$Z_{\mathrm{sh}} = \int \beta_y(G(x,w))q_x(x;\eta_x)q_w(w;\eta_w)\,dx\,dw. \tag{46}$$

Natural-gradient moment increments obtained by differentiating $\log Z_{\mathrm{sh}}$ with respect to the moments of q_x and q_w are exactly the EP moment increments for the tilted distribution proportional to $\beta_y(G(x,w))q_x(x)q_w(w)$. The cavity q_x appears once in Eq. (46)*, regardless of how many times x is used inside G.*

Proof. Define the downstream message for x by $\beta_x(x) = \int \beta_y(G(x,w))q_w(w;\eta_w)\,dw$. Then $Z_{\mathrm{sh}} = \int \beta_x(x)q_x(x;\eta_x)\,dx$. Theorem 3.2 gives $\Delta\mu_x = F_x\nabla_{\mu_x}\log Z_{\mathrm{sh}} = \mathbb{E}_{\widehat{p}_{\mathrm{sh}}}[T_x(x)] - \mathbb{E}_{q_x}[T_x(x)]$, which is the EP moment increment for x. The same argument applies to w. Repeated internal uses of x affect the derivative of the deterministic map G in the vector-Jacobian product, but do not create extra factors of q_x in the tilted normalizer. □

Proposition B.5 (Duplicated cavity factors define a different tilted normalizer)**.** *Suppose a graph fragment uses one activation x in K branches, but an implementation introduces branch variables $x_1,\dots,x_K$, attaches a copy of the cavity density q_x to each branch, and then enforces equality with deterministic constraints. The copied-branch normalizer is*

$$Z_{\mathrm{copy}} = \int \beta_y(G(x_1,\dots,x_K,w))\left(\prod_{k=1}^{K} q_x(x_k)\right) q_w(w)\prod_{k=2}^{K}\delta(x_k - x_1)\,dx_{1:K}\,dw. \tag{47}$$

After integrating the equality constraints,

$$Z_{\mathrm{copy}} = \int \beta_y(G(x,\dots,x,w))q_x(x)^K q_w(w)\,dx\,dw. \tag{48}$$

This differs from the shared-variable normalizer, which contains one factor of $q_x(x)$. Thus duplicated cavity factors generally change the local tilted distribution unless the extra factors are removed by the site or region-graph accounting.

Proof. Equation (48) follows directly from Eq. (47) by integrating $x_2,\dots,x_K$ against the equality constraints $\delta(x_k - x_1)$. The correct shared-variable construction integrates over one variable x and therefore contains one cavity factor $q_x(x)$, as in Eq. (46). The extra density power generally changes both the normalizer and the tilted distribution. □

For a Gaussian cavity $q_x = \mathcal{N}(m,V)$, the product $q_x(x)^2$, after normalization, is proportional to $\mathcal{N}(m,V/2)$. Thus a copied residual branch with two equalized copies changes the covariance even before any observation is applied. Proposition B.4 avoids this by representing the residual or concatenation as a deterministic map of one shared moment state.

Duplicating a density and discarding cross-covariances are separate operations. One shared belief avoids the extra cavity factor, but a diagonal projection can still discard dependence between its branches. The following moment maps retain that dependence before projection.

Corollary B.6 (Residual and concatenation moment maps)**.** *Consider a residual block $y = Sx + F(x,w)$ with independent input and parameter blocks x, w and first-order local approximation $F(x,w) \approx F(m_x,m_w) + J_x(x-m_x) + J_w(w-m_w) + \epsilon$, where ϵ has zero mean, has covariance V_ϵ and is independent of x, w. The shared-variable moment map is*

$$m_y = Sm_x + F(m_x,m_w), \qquad V_y \approx (S+J_x)V_x(S+J_x)^\top + J_wV_wJ_w^\top + V_\epsilon. \tag{49}$$

For a concatenation $y = [x, F(x,w)]$, the corresponding first-order covariance is

$$V_y \approx \begin{pmatrix} V_x & V_xJ_x^\top \\ J_xV_x & J_xV_xJ_x^\top + J_wV_wJ_w^\top + V_\epsilon \end{pmatrix}. \tag{50}$$

A diagonal implementation retains the diagonal of these covariances, or fuses the concatenation with the next affine module before diagonal projection.

Proof. Both formulas are the first and second moments of the shared-variable deterministic maps under the local linear approximation. In the residual case, the perturbation in x enters through the composite sensitivity $S + J_x$, giving Eq. (49). In the concatenation case, the first block is x and the second block is the linearized residual branch, so the cross-covariance between the two blocks is $\mathrm{Cov}(x, J_x x) = V_x J_x^\top$. These are exact moments of the affine surrogate. Proposition B.2 supplies its backward corrections; their EP interpretation requires the corresponding fixed-message normalizer condition. □

For a compact backward rule, hold the coefficients and intercept of the local affine expansion, and V_ϵ, fixed. Write $A = S + J_x$, $B = J_w$ for the residual block. If $r = \partial \ell / \partial m_y$ and $H = \partial \ell / \partial V_y = H^\top$, then

$$g_{m_x} = A^\top r, \quad g_{V_x} = A^\top H A, \qquad g_{m_w} = B^\top r, \quad g_{V_w} = B^\top H B. \tag{51}$$

These follow by differentiating the affine mean and covariance maps. For concatenation, use $A = [I; J_x]$ and $B = [0; J_w]$ in the same rule. The Gaussian conversions in Appendix B.5 then encode the adjoints. The residual covariance includes $S V_x J_x^\top + J_x V_x S^\top$; simply adding independent skip and branch variances would omit this contribution, including its diagonal. If a moment map differentiates its linearization matrices as functions of the input moments, their additional derivatives must also be included.

B.5 Gaussian covariance blocks and EP updates

Gaussian adjoint conversion. For $q = \mathcal{N}(m, C)$ with $C \succ 0$, use raw moments $\mu = (m, Q)$, $Q = C + mm^\top$, and canonical parameters $(h, -\Lambda/2)$, where $\Lambda = C^{-1}$ and $h = \Lambda m$. Symmetric-matrix derivatives use the trace inner product and nonredundant symmetric coordinates. For a scalar $\ell(m, C)$, let $a = \partial \ell / \partial m|_C$ and $G = \partial \ell / \partial C = G^\top$. The Gaussian signal and its inverse conversion are

$$d_m = Ca, \qquad d_Q = m d_m^\top + d_m m^\top + 2CGC, \tag{52}$$

$$a = C^{-1} d_m, \qquad G = \tfrac{1}{2} C^{-1} (d_Q - m d_m^\top - d_m m^\top) C^{-1}. \tag{53}$$

Derivation. The raw-moment adjoint is $(a - 2Gm, G)$. A canonical perturbation $(\delta h, \delta(-\Lambda/2)) = (b, H)$ gives $\delta C = 2CHC$ and $\delta m = C(b + 2Hm)$. Substituting $(b, H) = (a - 2Gm, G)$ and differentiating $Q = C + mm^\top$ gives Eq. (52); solving for a, G gives Eq. (53). The diagonal cases recover Eqs. (3)–(4).

Finite EP moment update. For moment-damped EP, Eq. (5) applies to raw moments. For valid Gaussian endpoints, its covariance satisfies

$$C^+ = (1 - \rho) C + \rho C^\star + \rho(1 - \rho)(m^\star - m)(m^\star - m)^\top, \qquad 0 \le \rho \le 1, \tag{54}$$

as follows by expanding $Q^+ - (m^+)(m^+)^\top$. This preserves covariance validity. General approximate directions still require a valid-domain check. The shared reverse transport therefore supports distinct finite EP and MFVI updates.

B.6 Affine propagation with correlated weights and activations

Retaining covariance blocks changes the forward moments as well as their backward conversion. The following total-covariance calculation provides the affine rule; correlated-weight moment propagation is also developed in Li et al. [22].

Proposition B.7 (Correlated affine moments)**.** *Let $z = Wa + b$, with mutually independent blocks W, a, b and finite second moments. Write $M = \mathbb{E}[W]$, $m_a = \mathbb{E}[a]$, $m_b = \mathbb{E}[b]$, $C_a = \mathrm{Cov}(a)$, $C_b = \mathrm{Cov}(b)$, and $\Omega_{ij,k\ell} = \mathrm{Cov}(W_{ij}, W_{k\ell})$. Then*

$$m_z = M m_a + m_b, \qquad C_z = \mathbb{E}_W[W C_a W^\top] + \mathrm{Cov}_W(W m_a) + C_b, \tag{55}$$

$$(C_z)_{ik} = (M C_a M^\top)_{ik} + (C_b)_{ik} + \sum_{j,\ell} \Omega_{ij,k\ell} \big((C_a)_{j\ell} + (m_a)_j (m_a)_\ell \big). \tag{56}$$

These moments allow correlations within both the activation block and the weight block. They do not require Gaussian block distributions.

Proof. Independence gives $\mathbb{E}[z \mid W] = Wm_a + m_b$ and $\mathrm{Cov}(z \mid W) = WC_aW^\top + C_b$. Taking expectations and applying the law of total covariance proves Eq. (55). Expanding $\mathbb{E}[W_{ij}W_{k\ell}] = M_{ij}M_{k\ell} + \Omega_{ij,k\ell}$ in its two weight-dependent terms gives Eq. (56). □

Corollary B.8 (Backward rule for correlated affine moments)**.** *For the moment map above, let* $\alpha = \partial\ell/\partial m_z|_{C_z}$ *and* $G = \partial\ell/\partial C_z = G^\top$. *Define*

$$S_a = C_a + m_a m_a^\top, \qquad (K_G)_{j\ell} = \sum_{i,k} G_{ik}\Omega_{ij,k\ell}.$$

Then the stored-coordinate adjoints are

$$\begin{aligned} g_{m_a} &= M^\top\alpha + 2K_G m_a, & g_{C_a} &= M^\top GM + K_G, \\ g_M &= \alpha m_a^\top + 2GMC_a, & (g_\Omega)_{ij,k\ell} &= G_{ik}(S_a)_{j\ell}, \\ g_{m_b} &= \alpha, & g_{C_b} &= G. \end{aligned} \tag{57}$$

Gaussian conversion by Eq. (52) *encodes each pair as a backward pseudo-gradient. For* W*, use its vectorized mean and covariance block.*

Proof. Differentiate Eq. (56). The weight-covariance term contributes $\mathrm{tr}(K_G\, dC_a) + 2(K_G m_a)^\top dm_a$ and has Ω-adjoint $G_{ik}(S_a)_{j\ell}$. Differentiating $Mm_a + m_b$ and $MC_aM^\top + C_b$ gives the remaining terms. Covariance symmetry and $G = G^\top$ imply $K_G = K_G^\top$. Proposition B.2 and the Gaussian conversion complete the reverse calculation. □

Covariance adjoints use the symmetric subspace and trace inner product. The forward covariance is positive semidefinite for valid input and weight covariances; Fisher inversions and belief updates use positive definite Gaussian blocks and must remain in that domain. Deterministic or singular intermediate states carry ordinary adjoints or use fixed-support coordinates; they are not decoded by inverting the Fisher matrix of the full singular block. For EP, the terminal scalar is the stated fixed-message log normalizer. Proposition B.2 converts its moment-map derivatives into the corresponding belief corrections. Appendix C uses the same covariance derivatives for the expected log likelihood and supplies the variational prior, entropy, and finite canonical update.

Diagonal C_a and independent weight entries recover the marginal variance rule in Eq. (6). Keeping covariance entries lets the first term propagate activation correlations and the weight covariance terms propagate correlated parameter uncertainty. Differentiating these formulas, followed by Eqs. (41) and (52), constructs the corresponding backward pseudo-gradients.

Although these first two moments are exact under the stated assumptions, Wa is generally non-Gaussian. Representing its output by a Gaussian is a moment projection. Dependence between W and a requires joint mixed moments beyond the independent-block formula. A covariance representation must also specify which blocks are carried through subsequent modules: correlations discarded in the forward pass cannot be restored by reverse differentiation. These choices determine both the approximation and its storage and computation costs.

B.7 Checks for literal implementations

The statements above lead to direct numerical checks. Each check uses its stated fixed-message normalizer. First, the diagonal Gaussian conversions in Eqs. (38)–(39) should be inverse maps for random positive variances. Second, on a linear-Gaussian chain, full Gaussian moment backpropagation should match exact Kalman/RTS smoothing, and a diagonal projection of the resulting tilted Gaussian should keep the diagonal posterior covariance rather than the inverse diagonal precision. Third, for an affine Gaussian block $y = Ax + Bw + b + \epsilon$, the natural-gradient moment update for w should match the literal Gaussian EP tilted marginal. Fourth, for a residual block $y = Sx + Ax + Bw + b + \epsilon$, the shared-variable moment map in Eq. (49) should match literal EP, while the copied-branch normalizer in Eq. (48) should show the extra cavity factor. Finally, for concatenation, the shared-variable check is the cross-covariance block $V_x J_x^\top$ in Eq. (50). These checks are small Gaussian problems and are independent of the neural-network implementation.

C Mean-field Variational Inference

This appendix derives the block mean-field variational objective, its backward pseudo-gradient, and its finite belief update. The module reverse rule is Proposition B.2: here its terminal scalar is the expected log likelihood. The Gaussian signal conversions and correlated moment maps are stated in Appendix B; the prior and entropy contributions and the canonical update are specific to the variational objective below.

For each regular minimal factor, write $\mu_u = \nabla A_u(\eta_u)$ and $F_u = \partial\mu_u/\partial\eta_u$. We work at interior beliefs with invertible Fisher blocks, as in Appendix B.

C.1 Block mean-field variational learning

Let $q(W) = \prod_u q_u(W_u; \eta_u)$ and $p_0(W) = \prod_u p_{0,u}(W_u; \eta_{0,u})$, with each factor and its prior in the same exponential family. Variables within a block may be correlated. Define the ELBO

$$\mathcal{L}(\mu) = \Psi(\mu) - \sum_u \mathrm{KL}(q_u \| p_{0,u}), \qquad \Psi(\mu) = \sum_{n=1}^{N} \mathbb{E}_q[\log p(y_n \mid x_n, W)]. \tag{58}$$

The variational factors are the parameter beliefs $q_u(W_u)$. Activation moments are induced intermediate states: their backward signals encode adjoints and introduce no additional variational factors or KL terms. Persistent variational updates act on parameter beliefs. The following is the standard variational natural-gradient/mirror update [16], expressed through the module backward rule.

Proposition C.1 (Variational pseudo-gradients and belief updates)**.** *Assume the expectations and derivatives in Eq.* (58) *exist. The variational adjoint and its moment-space signal are*

$$g_u^{\mathrm{VI}} = \nabla_{\mu_u}\Psi + \eta_{0,u} - \eta_u, \qquad d_u^{\mathrm{VI}} = F_u g_u^{\mathrm{VI}}. \tag{59}$$

If the forward moment computation evaluates Ψ, Eq. (41) *computes its likelihood adjoints. A computation of $\widehat{\Psi}$ instead gives the corresponding surrogate adjoints. For likelihood-only adjoints, add the prior and entropy contribution $\eta_{0,u} - \eta_u$ once at each parameter belief. If the reverse pass already differentiates the full variational objective, no further prior term is added. For $\rho > 0$, the persistent belief takes the canonical update*

$$\eta_u^+ = \eta_u + \rho g_u^{\mathrm{VI}} = (1-\rho)\eta_u + \rho\big(\eta_{0,u} + \nabla_{\mu_u}\Psi\big), \qquad \mu_u^+ = \nabla A_u(\eta_u^+), \tag{60}$$

provided η_u^+ lies in the natural-parameter domain.

Proof. Writing the KL divergence as $(\eta_u - \eta_{0,u})^\top \mu_u - A_u(\eta_u) + A_u(\eta_{0,u})$ and differentiating in mean coordinates gives $\nabla_{\mu_u}\mathrm{KL} = \eta_u - \eta_{0,u}$. This proves Eq. (59). The mirror ascent step, with convex dual A_u^* as its mirror potential, has stationarity equation $g_u^{\mathrm{VI}} - \rho^{-1}(\eta_u^+ - \eta_u) = 0$, giving Eq. (60). Equivalently, $F_u^{-1}\nabla_{\eta_u}\mathcal{L} = \nabla_{\mu_u}\mathcal{L}$ is the canonical natural-gradient direction. Proposition B.2 supplies the likelihood adjoints. □

Corollary C.2 (Variational messages and pseudo-gradients)**.** *Write the likelihood as $\prod_r f_r(W)$ and evaluate all derivatives at the current variational state. Define*

$$\lambda_{r\to u} = \nabla_{\mu_u}\mathbb{E}_q[\log f_r(W)], \qquad \eta_u^{\mathrm{tar}} = \eta_{0,u} + \sum_r \lambda_{r\to u}. \tag{61}$$

When the forward computation evaluates these expected log factors, its reverse computation supplies the contributions. The signal and update are

$$d_u^{\mathrm{VI}} = F_u(\eta_u^{\mathrm{tar}} - \eta_u), \qquad \eta_u^+ = \eta_u + \rho F_u^{-1} d_u^{\mathrm{VI}}, \qquad \mu_u^+ = \nabla A_u(\eta_u^+). \tag{62}$$

If, with the other variational factors fixed,

$$\mathbb{E}_{q_{-u}}[\log f_r(W)] = c_{r,u} + \langle \lambda_{r\to u}, T_u(W_u)\rangle, \tag{63}$$

then these coefficients are the exact conjugate variational messages. For an admissible target, $\rho = 1$ gives the optimal coordinate update of this one block with all other factors fixed.

Proof. Linearity gives $\nabla_{\mu_u}\Psi = \sum_r \lambda_{r\to u}$, so Proposition C.1 gives the signal and update. Under Eq. (63), the unrestricted mean-field coordinate optimum is

$$q_u^\star(W_u) \propto p_{0,u}(W_u)\exp\left\{\sum_r \mathbb{E}_{q_{-u}}[\log f_r(W)]\right\}.$$

Its canonical parameter is η_u^{tar}, which is attained by the unit step. Differentiating the expected log factors through their moment computation gives their coefficients by Proposition B.2. □

This is the conjugate VMP coordinate update [45]. Outside the conjugate case, the derivatives $\lambda_{r\to u}$ define local CVI/mirror contributions [16]; they need not be canonical coefficients of the exact factor message $\exp\{\mathbb{E}_{q_{-u}}[\log f_r]\}$. The transported d_u^{VI} is a local mean-coordinate direction; the finite change $\mu_u^+ - \mu_u$ generally differs from ρd_u^{VI}.

For uniform minibatch sampling, Ψ can be estimated by $N/|B|$ times its sum over minibatch B; the KL contribution enters once per parameter belief, including a belief reused by several modules. This estimates one fixed dataset objective across epochs. If the forward computation replaces Ψ by a moment approximation $\widehat{\Psi}$, the same derivation applies to $\widehat{\Psi} - \sum_u \mathrm{KL}(q_u\|p_{0,u})$; that surrogate is not automatically an evidence lower bound for the original model.

C.2 Gaussian variational updates

Use the Gaussian coordinates and adjoint conversion in Appendix B.5.

Finite Gaussian variational update. For the variational update, now take $a = \partial\Psi/\partial m|_C$ and $G = \partial\Psi/\partial C$. With Gaussian prior (m_0, C_0), set $\Lambda_0 = C_0^{-1}$ and $h_0 = \Lambda_0 m_0$. Substituting the raw-moment adjoint into Eq. (60) yields

$$\Lambda^+ = (1-\rho)\Lambda + \rho(\Lambda_0 - 2G), \qquad h^+ = (1-\rho)h + \rho(h_0 + a - 2Gm), \tag{64}$$

$$C^+ = (\Lambda^+)^{-1}, \qquad m^+ = C^+h^+ = m + \rho C^+\big[a - \Lambda_0(m - m_0)\big]. \tag{65}$$

These formulas require $\Lambda^+ \succ 0$; the finite mean update uses the new covariance C^+. They implement a canonical mirror step.

Gaussian observation example. Consider $y = w^\top a + \epsilon$, with independent w, a and independent noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 > 0$. Hold the input distribution fixed while updating $q(w)$; this input may be induced by upstream parameter beliefs. Set $S_a = C_a + m_a m_a^\top$ and $Q_w = C_w + m_w m_w^\top$. The exact expected log likelihood is

$$\Psi = -\frac{y^2 - 2y\, m_w^\top m_a + \mathrm{tr}(Q_w S_a)}{2\sigma^2} + \mathrm{const.}$$

Its raw-moment derivatives are $(ym_a/\sigma^2, -S_a/(2\sigma^2))$, giving the conjugate target

$$\Lambda_w^{\mathrm{tar}} = \Lambda_{0,w} + S_a/\sigma^2, \qquad h_w^{\mathrm{tar}} = h_{0,w} + ym_a/\sigma^2.$$

Although $w^\top a$ can be non-Gaussian, its first two moments evaluate this quadratic expected log likelihood exactly. The formulas above therefore recover the conjugate coordinate update, including the input correlations in S_a.

Correlations within variational blocks. To retain correlations within activation and weight blocks, use Proposition B.7 and Corollary B.8 to evaluate and differentiate the moment graph, with Ψ as the terminal scalar. Add the local prior and entropy contribution $\eta_0 - \eta$ and apply Proposition C.1's canonical update. Different variational weight blocks remain factorized, but entries within a block need not be independent.

D Experiment Details

Our experiments were conducted in Google Colab notebooks using the R `IRkernel` together with TensorFlow/Keras and TensorFlow Probability through Python interop. We used NVIDIA T4, L4, and A100 GPUs depending on the experiment size. The supplementary tutorial provides executable examples with environment setup, dataset loading, seeds, training and evaluation code, and separate implementation checks.

All datasets and pretrained assets used in the experiments are public research benchmarks or public model checkpoints cited in the main text. The supplementary notebooks include links to the official sources, note the applicable dataset/model licenses or terms of use when provided by the source, and avoid redistributing restricted raw data. New code introduced by this work is documented in the notebooks and released with instructions for reproducing the tutorial's examples and outputs.

D.1 Performance, Calibration, and Efficiency

This appendix provides detailed experimental procedures supporting our evaluation of Bayesify with projected expectation propagation (EP). We systematically compare Bayesify EP against established Bayesian and uncertainty-aware baselines, focusing on three key aspects:

1. **Predictive Performance on In-Distribution (ID) Data:** Can the model reliably and accurately predict outcomes on test data drawn from the same distribution as the training data?
2. **Uncertainty Calibration and Out-of-Distribution (OOD) Detection:** Do the model's predicted probabilities accurately represent its true confidence? Additionally, when faced with novel or unexpected inputs, can the model effectively detect and indicate lower confidence?
3. **Computational Efficiency:** What computational resources are required at inference and during training to achieve robust uncertainty quantification? Specifically, we measure inference-time overhead (e.g., number of forward passes, memory usage, latency) and training-step overhead.

Our central hypothesis is that Bayesify EP achieves calibration and OOD robustness comparable to computationally expensive methods such as deep ensembles, but crucially, it requires only a single forward moment pass per prediction at inference. Propagating activation moments and maintaining parameter beliefs incur additional computation and memory; Table 4 reports the measured WRN-28-10 overhead. The subsequent sections document the experimental setup and evaluation protocols.

Datasets and Tasks. We evaluate Bayesify EP across widely adopted benchmarks spanning both vision and NLP tasks, selected to rigorously assess predictive accuracy and robustness in uncertainty estimation.

For vision, we use **CIFAR-10**, containing 50k training and 10k test images (32×32 RGB; 10 classes). We hold out 5k images from the CIFAR-10 training set as a dev set for early stopping and temperature fitting, training on the remaining 45k. Deterministic baselines are evaluated at their best-dev-NLL checkpoint; ensembles use the best-dev-NLL checkpoint for each member; Laplace freezes the best-dev feature extractor; MC-Dropout reuses the same checkpoint with dropout active at test time; SWAG uses a late-training weight collection as described below. To test robustness under distribution shift, we employ **CIFAR-10-C**, which applies multiple corruption types at five severities to the CIFAR-10 test set (we report macro-averages across all 15×5 corruption types and severities). We measure near-OOD detection performance using **CIFAR-100**, comprising 50k training and 10k test images (100 classes), concatenated with CIFAR-10 test for AUROC evaluation (ID=0 for CIFAR-10, OOD=1 for CIFAR-100).

Additionally, we evaluate on the **Camelyon17-WILDS** dataset, a histopathology benchmark designed explicitly to assess robustness to domain shifts. It includes roughly 302k training, 34.9k validation, and 85k test patches (each 96×96) extracted from whole-slide images collected across five hospitals. Following the WILDS protocol, a patch is labeled positive if the *central* 32×32 region overlaps any tumor annotation; otherwise negative. We adopt the official domain split: training on three hospitals, validating on one (OOD), and testing on a held-out hospital (OOD). We additionally hold

out an *in-distribution* dev split from the training-hospital data; this split is used for early stopping and temperature scaling. The WILDS val/test hospitals are not used for model selection.

For NLP, we select the **CLINC Out-of-Scope (OOS)** intent classification dataset, constructed to emulate realistic conversational AI use-cases. It contains 150 in-scope intents across 10 domains, totaling 22,500 in-scope utterances split as 100/20/30 per intent for train/validation/test, plus 1,200 out-of-scope utterances split 100/100/1,000 for train/val/test. ID metrics are computed on in-scope classes; OOD detection uses the OOS split scored by predictive entropy. These benchmarks are widely recognized within the uncertainty quantification community, reflecting realistic deployment scenarios where predictive reliability and calibrated confidence estimates are essential.

Architectures and Training Protocols. We choose well-established architectures and standard training protocols, reflecting commonly adopted best practices. All experiments run on TensorFlow ≥ 2.14 with a matching TensorFlow Probability version. We set `tf.random.set_seed`, `np.random.seed`, and Python `random.seed` to the same value and repeat the full pipeline over three distinct seeds (0/1/2); unless noted, we enable mixed precision with `tf.keras.mixed_precision.set_global_policy(''mixed_float16'')` while casting loss computations to float32 to avoid numeric drift. The tables report means over the three runs.

For CIFAR-based datasets (CIFAR-10, CIFAR-10-C, CIFAR-100), we utilize the **Wide ResNet-28-10 (WRN-28-10)**, favored for its robust baseline performance in uncertainty studies. Training uses SGD with momentum (0.9), batch size 128, weight decay (5×10^{-4}), label smoothing 0.1, and either cosine-annealed learning rates over 200 epochs or a step schedule (base LR 0.1 with drops at epochs 60/120/160). Data augmentation follows standard practice: per-image standardization, random crop with 4-pixel padding to 32×32, and random horizontal flip; evaluation uses a center crop with no flip. For CIFAR-10-C, no train-time augmentation is applied; we reuse the same checkpoint for all corruption types and severities.

For the Camelyon17-WILDS dataset, we adopt **DenseNet-121**, a widely-used architecture in medical imaging due to its efficiency and effectiveness. The tf.data pipeline resizes patches to 224×224; training uses light color jitter and flips, and evaluation uses center-crop/resize only. Optimization uses SGD or AdamW (both acceptable here), moderate weight decay (1e-4 to 5e-4), and a multi-step learning rate schedule, with early stopping based on ID-dev NLL (held out from train hospitals). Temperature scaling is fit only on this ID-dev split.

For the CLINC OOS NLP task, we fine-tune a pretrained **RoBERTa-base** transformer model. We follow RoBERTa defaults: byte-level BPE tokenizer with no lowercasing, fixed normalization, and maximum sequence length chosen to cover >99% of utterances without truncation (typically 128–256). Fine-tuning uses AdamW (lr 2e-5, weight decay 0.01), up to 10 epochs with early stopping on dev NLL, and gradient clipping at a fixed global norm of 1.0.

Baseline Methods and Implementations. We benchmark Bayesify EP against representative uncertainty estimation methods chosen to cover major categories prevalent in the Bayesian deep learning literature, implemented in the TensorFlow ecosystem with shared training budgets (epochs × images/step) and identical augmentations. Methods that require test-time sampling report and *incur* their exact Equivalent Forward Passes (EFP) at inference. Hyperparameters are tuned on the ID dev set via small grids around community defaults; no corrupted/OOD data are used for tuning.

- **Deterministic + Temperature Scaling (Det+TS):** Train deterministically; freeze the network and fit a single scalar temperature $T>0$ on the CIFAR-10 dev set by minimizing mean NLL with Adam (lr 0.05) for 200–500 steps, parameterizing $T = \mathrm{softplus}(\tau) + \epsilon$ to enforce positivity. Reuse the fitted T unchanged for all evaluations (ID, CIFAR-10-C, OOD).
- **Deep Ensembles (Ens(5)):** Train five independent WRN-28-10 models with different seeds/data orders using the exact recipe above; average softmax probabilities at test time.
- **MC-Dropout:** Insert dropout in residual/conv blocks and before the classifier (drop rate tuned on dev, e.g., 0.1–0.3). Keep dropout active at test time; draw $S=20$ stochastic forward passes and average probabilities.
- **Bayes-by-Backprop (BBB):** Replace weight-bearing layers with TFP reparameterized convolutional and dense variants; keep batch-norm statistics deterministic. Optimize the

ELBO with the reparameterization trick and KL annealing (KL weight linearly annealed from 0 to 1 over the first 30% of epochs); at test time, draw S=10–20 full-network samples and average.

- **SWAG(20):** After burn-in, periodically collect full-network weights over the last 30–50 epochs, estimate a low-rank+diagonal Gaussian posterior (running mean and covariance), and sample S=20 weight draws at test time; optionally report a classifier-head-only ablation.
- **Laplace (last-layer):** Freeze the feature extractor at the best-dev-NLL checkpoint; form a diagonal likelihood-curvature approximation F for the last layer. An empirical-Fisher diagonal averages squared per-example NLL gradients; squaring a minibatch-mean gradient gives a different approximation. Form the posterior precision $\Lambda = \lambda I + F$ (using training weight decay as a prior proxy) and sample S=20 from $\mathcal{N}(w^*, \Lambda^{-1})$ with TFP; average probabilities.
- **SNGP/DUE (single-pass):** Spectral normalization with GP-inspired heads (random Fourier features or inducing points). Single forward pass at test time (EFP=1).
- **Ours (Bayesify EP, single-pass):** Selected Dense/Conv kernels and RoBERTa parameter blocks carry diagonal-Gaussian beliefs; unselected parameters remain point-valued. The forward pass propagates activation moments. The reverse pass supplies corrections to persistent parameter beliefs, which are updated by Eq. (5). Shared activations retain one state, and corrections from their uses accumulate before an update. The schedule stores no per-example EP sites.

Training messages and predictive readout. During training, the Gaussianized classification head in Algorithm 1 conditions the output logit moments on the label. An absolute target $(m^\star, v^\star)$ corresponds to the raw-moment corrections $\Delta m = m^\star - m$ and $\Delta s = v^\star + (m^\star)^{\odot 2} - (v + m^{\odot 2})$. Appendix B gives their conversion to module adjoints. The reverse pass accumulates parameter corrections, combines the normalized observation and fixed-prior contributions, and applies the damped update in (m, s) coordinates once per persistent belief. Prediction uses only the forward moment pass: binary tasks use a logistic-probit readout, and multiclass tasks use ID-dev-calibrated variance-aware logit scaling, fixed across ID/OOD evaluation.

Metrics and Evaluation Protocol. We use standard metrics, separately evaluating in-distribution prediction accuracy and uncertainty robustness under distribution shifts. **Accuracy (Acc↑)** is the fraction of correct predictions. **Negative Log-Likelihood (NLL↓)** reflects probability quality, with low values for correct confident predictions and large penalties for confident errors. **Expected Calibration Error (ECE↓)** measures alignment between predicted confidence and empirical accuracy via M equal-width bins (default M=15): $\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$. For OOD detection, we score examples by predictive entropy $\mathcal{H}\big(p_\theta(y\,|\,x)\big) = -\sum_c p_\theta(y{=}c\,|\,x) \log p_\theta(y{=}c\,|\,x)$, and compute **AUROC↑** by ranking (higher entropy = more OOD-like). On CIFAR-10-C, we report **cNLL↓** and **cECE↓** by computing NLL/ECE per corruption and severity and macro-averaging across all conditions. Temperature scaling is fit exclusively on the ID dev set and reused unchanged across all evaluations.

D.2 LLMs & PEFT: Bayesian adapters without multi-sampling

This appendix details the experimental protocol for parameter-efficient fine-tuning (PEFT) with LoRA adapters on the `gpt2_base_en` backbone, evaluating Bayesian surrogates that avoid multi-sample inference. We compare single-pass Bayesify EP adapters to practical baselines (Det+TS, Laplace, mean-field VI) on ID tasks (SST-2, RTE) and OOD detection via paired corpora (IMDb for SST-2; HANS for RTE). The goal is to test whether adapter-level Bayesian updates can deliver calibrated uncertainty and OOD robustness without test-time sampling.

Environment and Seeding. Experiments run in Colab with TensorFlow $\geq$2.14, TensorFlow Probability matched to TF, KerasNLP for the `gpt2_base_en` preset, and Hugging Face `datasets`. We set `tf.random.set_seed`, `np.random.seed`, and Python `random.seed` to the same value and repeat all experiments with three seeds (0/1/2). Mixed precision is enabled via `tf.keras.mixed_precision.set_global_policy(''mixed_float16'')` while cast-

ing loss computations to float32; disabling mixed precision increases determinism at the cost of speed.

Datasets and Tasks. We fine-tune on **SST-2** and **RTE** and use their paired out-of-distribution sets **IMDb** (for SST-2) and **HANS** (for RTE). Tokenization uses the `gpt2_base_en` byte-level BPE tokenizer (no lowercasing). We fix maximum sequence lengths at 256 for SST-2/IMDb and 384 for RTE/HANS; truncation/padding follow the model preset defaults. The table reports accuracy separately for SST-2 and RTE and averages NLL/ECE across the two tasks. OOD AUROC uses the paired concatenation (ID test + OOD partner).

Model and PEFT Setup. We instantiate the `gpt2_base_en` backbone with a sequence-classification head and insert LoRA adapters into the attention projection Dense layers. In implementations with fused query/key/value projections, the adapters are applied to the corresponding query/value slices or equivalent projection sublayers. Using column-vector inputs as in the main affine rule, a wrapped weight $W \in \mathbf{R}^{d_{\text{out}} \times d}$ has low-rank factors $A \in \mathbf{R}^{r \times d}$ and $B \in \mathbf{R}^{d_{\text{out}} \times r}$. The rank is r=16, and the adapter multiplier is α/r. Optional LoRA dropout p=0.05 is applied to the adapter input. The adapted linear is $y = Wx + \frac{\alpha}{r} B(A(\text{dropout}(x)))$. The parameter mean initializations are $M_B = 0$ and entries of M_A drawn from $\mathcal{N}(0, 0.02)$. For Bayesian adapters, posterior variances are separate belief parameters. LoRA dropout (p=0.05) is applied only during training and is disabled at evaluation. We *freeze the backbone* and train *only* LoRA parameters and the classification head. LoRA adapters are implemented as custom Keras layers for TensorFlow.

Training Protocol. The gradient-based baselines use AdamW (learning rate 2×10^{-4}, weight decay 0.01), with gradient clipping at 1.0. Training uses batch size 32 and up to 5 epochs with early stopping on dev NLL. We save each method's best-dev-NLL checkpoint. Laplace uses that method's MAP checkpoint to form its posterior; Det, MFVI, and our Bayesify EP adapters are evaluated at their own best-dev-NLL checkpoints. Tokenization/batching are implemented with `datasets` and HF TF data collators. AdamW weight decay is applied to trainable weights except biases and LayerNorm parameters (no decay on those).

Baselines and Implementations. We standardize training budgets across baselines. Test-time sampling methods *incur and report* their Equivalent Forward Passes (EFP). Hyperparameters are tuned on ID dev sets via small grids around community defaults; no OOD data are used for tuning.

- **Deterministic + Temperature Scaling (Det+TS):** After fine-tuning, fit a single scalar temperature $T>0$ on each task's dev logits by minimizing mean NLL (Adam, lr 0.05, 200–500 steps; $T = \text{softplus}(\tau)+\epsilon$), keeping model weights fixed. Apply z/T at evaluation. TS is used *only* for this baseline. The temperature is fit on the ID dev set and reused unchanged for OOD/paired evaluations (no OOD refitting).
- **Laplace (last-layer/adapters):** At the MAP checkpoint, form a diagonal likelihood-curvature approximation F over all LoRA weights. For the empirical-Fisher convention, average squared per-example NLL gradients over examples once. Form $\Lambda = \lambda I + F$, where λ equals the L2 weight-decay coefficient used during training, and define $q(w) = \mathcal{N}(w^*, \Lambda^{-1})$. At test time, draw S=20 LoRA weight samples and average predictive probabilities (backbone frozen).
- **Mean-field VI (adapters only):** Parameterize each LoRA weight with $q(w_i) = \mathcal{N}(\mu_i, \sigma_i^2)$ and minimize the negative ELBO $\mathbf{E}_q[\text{NLL}] + \text{KL}(q\|p)$ via reparameterization gradients (TFP VI utilities). The prior is Gaussian with precision λ matched to the training weight decay (λ equals the L2 coefficient; prior std $= 1/\sqrt{\lambda}$). The KL weight is linearly warmed up from 0 to 1 over the first epoch. At test time, sample S=10 adapter draws and average probabilities (backbone frozen).
- **Ours (Bayesify EP adapters, single-pass):** Both LoRA factors A and B carry diagonal-Gaussian beliefs while the backbone remains frozen. The forward pass propagates moments through Ax and $B(Ax)$ and combines the adapter and frozen paths. The Gaussianized observation head supplies target logit moments during learning; custom reverse rules propagate the corresponding corrections to both factors. Their persistent first and second raw moments are updated by Eq. (5), with the fixed-prior contribution retained across minibatches and

epochs. The classification head remains deterministic. Prediction uses one forward moment pass and the binary readout described above, without label conditioning or weight samples.

Metrics and Evaluation Protocol. ID performance on SST-2 and RTE is measured with **Accuracy (Acc↑)**, **NLL↓** (mean cross-entropy), and **ECE↓** using $M{=}15$ equal-width confidence bins: $\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$. We report the average NLL/ECE across the two ID tasks. For **OOD AUROC**, we concatenate each ID test set with its OOD partner (SST-2→IMDb; RTE→HANS), label OOD=1, score by predictive entropy $\mathcal{H}(p) = -\sum_k p_k \log p_k$ (after TS for the Det+TS baseline only; the temperature is fit on ID dev and not refit on OOD), and compute AUROC by ranking entropy (higher = more OOD-like). For **AURC** (selective prediction), on each ID test set we sort by entropy *ascending* (lowest-uncertainty first), sweep coverage from 0 to 1 (101 points), keep the lowest-uncertainty items at each coverage, compute $\mathrm{risk} = 1 - \mathrm{accuracy}$ on the kept set, and integrate area via trapezoids (lower is better). All metrics are averaged over the three seeds, and we note Equivalent Forward Passes (EFP) at inference: ≈ 1 for Det+TS and our Bayesify EP adapters, and S for Laplace/VI sampling.

E Convex Duality, Variational Inference, and Reverse-Mode Automatic Differentiation

Convex conjugacy and exponential–family duality.

In Bayesian inference and optimization, we often need to move smoothly between two complementary viewpoints: the *canonical parameters*, which are directly adjusted during training or optimization, and the *mean parameters*, which represent the expectations we ultimately care about. Convex conjugacy provides an elegant mathematical tool to link these two perspectives. For a regular minimal exponential family, let its log-partition $A(\eta)$ be twice continuously differentiable on its open canonical domain Θ, with $\nabla^2 A \succ 0$. Work in nonredundant coordinates and the interior mean domain $\nabla A(\Theta)$. Its convex conjugate (or Fenchel dual) is defined by $A^*(\mu) = \sup_{\eta\in\Theta}\{\langle \eta, \mu\rangle - A(\eta)\}$. This duality provides a clean relationship between η and μ: by the Fenchel–Young inequality, we always have $\langle \eta, \mu\rangle \le A(\eta) + A^*(\mu)$, with equality precisely when the parameters are connected through the gradients $\mu = \nabla A(\eta)$ and $\eta = \nabla A^*(\mu)$. On these domains the gradient maps are inverses, giving a one-to-one correspondence between canonical and mean parameters. The local geometry around these parameters is captured by the Hessians $F(\eta) = \nabla^2 A(\eta) \succ 0$ and $F(\mu)^{-1} = \nabla^2 A^*(\mu) \succ 0$, often called Fisher information matrices. These matrices transform small perturbations between the two parameterizations, providing the local curvature for natural gradients and mirror updates. Finite-step stability also depends on the objective and update size.

This duality becomes particularly intuitive when we look at exponential-family distributions. Given sufficient statistics $T(z)$, a base measure $h(z)$, and canonical parameters η, the exponential family has a probability density $q_\eta(z) = h(z)\exp\{\langle \eta, T(z)\rangle - A(\eta)\}$, where $A(\eta) = \log \int h(z)\exp\{\langle \eta, T(z)\rangle\}dz$ ensures normalization. In this setting, the mean parameters $\mu = \mathbf{E}_{q_\eta}[T(z)]$ are exactly the gradient $\nabla A(\eta)$. Moreover, the Fisher information $F(\eta) = \mathrm{Cov}_{q_\eta}[T(z)]$ quantifies the sensitivity of mean parameters to changes in canonical parameters. Conversely, its inverse $F^{-1}(\mu)$ characterizes the sensitivity in the other direction. The exact entropy identity is $A^*(\mu) = -H(q_\mu) - \mathbf{E}_{q_\mu}[\log h(z)]$. A fixed base measure need not have a constant expectation under q_μ. With $h = 1$ relative to the chosen reference measure, A^* is exactly negative entropy. Thus, within any exponential family defined by the same sufficient statistics, the Kullback–Leibler divergence between two members simplifies neatly to the Bregman divergence: $\mathrm{KL}(q_\eta \| q_{\eta'}) = D_A(\eta' \| \eta)$ and $\mathrm{KL}(q_\mu \| q_{\mu'}) = D_{A^*}(\mu \| \mu')$. This link gives mirror descent and natural gradients the same local geometry. A natural gradient is the initial tangent of the corresponding mirror path; a finite mirror minimization is a separate update, as shown in Appendix A.

Let's ground these ideas in familiar examples. First, for a categorical distribution with parameter vector η, the log-partition is $A(\eta) = \log \sum_j e^{\eta_j}$, and mean parameters are simply probabilities $\mu = \mathrm{softmax}(\eta)$. Here, the Fisher matrix $F(\eta) = \mathrm{diag}(\mu) - \mu\mu^\top$ is singular along the constant direction (reflecting probability normalization), so we operate on the tangent space of probability vectors—directions preserving the sum-to-one constraint. Second, consider a Gaussian distribution $\mathcal{N}(m, \Sigma)$. In canonical (natural) form, parameters are given by $\eta_1 = \Sigma^{-1}m$, $\eta_2 = -\frac{1}{2}\Sigma^{-1}$, and mean parameters are $\mu = (m, M)$, where $M = \Sigma + mm^\top$. Perturbations between these parameterizations follow directly from the Fisher inverse: small changes in moments (dm, dM) map cleanly to canonical perturbations through $d\eta_2 = \frac{1}{2}\Sigma^{-1}(dM - m\,dm^\top - dm\,m^\top)\Sigma^{-1}$ and $d\eta_1 = \Sigma^{-1}dm - 2d\eta_2 m$. These explicit formulas are not arbitrary; they represent a local linearization that preserves the information geometry naturally encoded in the exponential family.

The central payoff of this perspective is conceptual clarity and practical simplicity. The dual coordinates η and μ provide global, interpretable parameterizations; the Hessians F and F^{-1} describe the local curvature for gradients and projections. The exact tree entropy is convex on its consistent mean domain under the assumptions of Appendix A; restricting a general variational objective to a mean-field family or introducing nonlinear likelihood expectations can still give nonconvex optimization. Ultimately, the duality framework equips us with robust, geometrically-informed tools to perform Bayesian inference, optimization, and uncertainty quantification in a principled and computationally convenient manner.

Variational inference and the Bethe free energy.

We have introduced exponential families and their dual parameterizations—natural parameters, mean parameters, and the dual potentials A and A^*. These tools simplify Bayesian inference by

giving explicit coordinate transformations between parameters and expectations, a built-in notion of curvature through the Fisher information, and clean geometric interpretations of natural gradients, mirror descent, and moment-matching projections. These advantages are not limited to small models; they become even more important in high-dimensional settings where direct inference is intractable: Many complex distributions exhibit conditional independence, and under strict positivity and the graph's Markov property, the Hammersley–Clifford theorem gives a factorization over its cliques. This factorization lets us break a massive joint problem into local pieces—cliques and separators—provided we maintain consistency between those pieces. In the next section we use this structured representation to formulate variational objectives and approximate inference schemes, and to show how natural gradients and moment matching emerge from message-passing algorithms on junction trees.

Variational inference (VI) leverages this structured perspective by formulating inference as an optimization problem. Recall that the exact posterior $p_\theta(z) \propto \exp\{\langle\theta, T(z)\rangle\}$ can be described through Gibbs' variational principle, $A(\theta) = \sup_q\{\mathbf{E}_q[\langle\theta, T(z)\rangle] + H(q)\}$, or equivalently as minimizing a KL divergence: $\min_q \mathrm{KL}(q\|p_\theta)$. Here the base measure is absorbed into the reference measure, and entropy is taken relative to that measure. However, this optimization is typically intractable because it requires handling expectations and entropy computations over all variables simultaneously. VI methods, therefore, introduce approximations that restore tractability, and these approximations fall broadly into two strategies.

The first strategy—commonly just called "variational inference"—simplifies the *approximate distribution itself*. We choose a simpler family $q(z) \in \mathcal{Q}_{\mathrm{VI}}$ whose conditional independence graph is sparse enough to allow tractable computations. For instance, the popular mean-field approximation completely removes interactions between variables, factorizing $q(z) = \prod_i q(z_i)$. More structured versions (such as tree-structured or low-treewidth VI) introduce a modest level of dependencies, preserving more fidelity to the original distribution but still ensuring computational feasibility. For a normalized target, $\mathcal{L}(q) = \mathbf{E}_q[\log p_\theta(z)] + H(q) = -\mathrm{KL}(q\|p_\theta)$, the ELBO up to the model evidence constant. A tractable entropy does not ensure tractable likelihood expectations; these may still require integration or approximation. Optimization typically involves iterative procedures such as coordinate ascent—updating each factor q_i given all others. The objective uses the reverse-KL direction $\mathrm{KL}(q\|p)$. It penalizes placing q-mass where p is small, but can underrepresent modes or uncertainty outside the mass retained by q.

The second strategy, exemplified by Bethe approximations, works with local beliefs and an approximate entropy, retaining only specified local consistency conditions. Specifically, we represent the posterior through a collection of local beliefs—clique marginals b_C and separator marginals b_S—and enforce only that these local marginals agree along their intersections (the separators). While the global entropy $H(q)$ remains intractable, Bethe approximation cleverly substitutes it with a sum of local entropies: each clique entropy is counted positively and each separator entropy negatively, capturing the graphical structure neatly. The resulting approximate objective, called the Bethe free energy, takes a simple form:

$$\Phi_{\mathrm{Bethe}}(\{\mu_C, \mu_S\}) = -\sum_{r\in\mathcal{C}\cup\mathcal{S}} \langle\theta_r, \mu_r\rangle + \sum_{C\in\mathcal{C}} A_C^*(\mu_C) - \sum_{S\in\mathcal{S}} A_S^*(\mu_S),$$

subject to affine consistency constraints that local means match on separators: $M_{C\to S}\mu_C + b_{CS} = \mu_S$. For full compatible region families on a junction tree, this entropy is exact and minimizing Φ_{Bethe} gives the posterior marginals. On loopy factor graphs, Bethe entropy uses the corresponding region counting numbers; its usual interior stationary points correspond to BP fixed points, without a general convexity or convergence guarantee. Tree-reweighted constructions can instead give convex surrogates.

Expectation propagation (EP) offers an algorithmic viewpoint of Bethe-like approximations. Rather than approximating a global entropy explicitly, EP maintains a set of simplified "site" approximations from a convenient exponential family. It iteratively refines these site approximations one at a time by matching a few chosen moments—essentially performing local "moment matching" steps. An EP update removes the old site contribution to form a cavity, combines it with the exact factor to form a tilted distribution $\widehat{p}$, and projects by minimizing $\mathrm{KL}(\widehat{p}\|q_\eta)$. Its derivative is $\nabla A(\eta) - \mathbf{E}_{\widehat{p}}[T]$, so an interior optimum matches the retained moments. This forward-KL projection differs from the VI objective above; the chosen family and approximation determine which dependence and uncertainty it retains.

Both of these strategies—global-structure VI and Bethe/EP approaches—share a common geometric and algebraic backbone, thanks to convex conjugacy. The relationships $\mu = \nabla A(\eta)$, $\eta = \nabla A^*(\mu)$, and the resulting Fisher matrices and KL–Bregman equivalences remain essential tools. They allow efficient computation of gradients, updates via natural-gradient steps (mirror descent), and straightforward moment-matching projections. Thus, while each variational approach differs in precisely *what* and *how* it approximates, both leverage the powerful duality structure provided by exponential families to make Bayesian inference practical even in complex graphical models.

Reverse–mode autodiff (VJP/JVP) and mirror maps.

Reverse-mode autodiff appears where our dual maps do their work: whenever a scalar objective depends on mean parameters μ or canonical parameters η and we want to send sensitivities back across the map $\mu = \nabla A(\eta)$ or $\eta = \nabla A^*(\mu)$. In this setting, μ and η are linked by the mirror maps of convex conjugacy, and the question reverse-mode answers is: given a derivative with respect to μ (or η), how can we efficiently and stably obtain the derivative with respect to η (or μ)?

It helps to picture the forward pass as choosing one side of the mirror and evaluating the quantities we need. If we work in natural parameters, we compute $\mu = \nabla A(\eta)$ and perhaps also the Fisher matrix $F(\eta) = \nabla^2 A(\eta)$ along the way (often implicitly via standard subroutines such as `logsumexp` for categoricals or Cholesky for Gaussians). Reverse-mode then pulls a covector $v = \partial L/\partial \mu$ back to η through the vector–Jacobian product $v \mapsto F(\eta)v$, because the Jacobian of μ with respect to η is precisely the Fisher information. Dually, if the forward pass worked in mean parameters and set $\eta = \nabla A^*(\mu)$, then reverse-mode propagates a covector $w = \partial L/\partial \eta$ back to μ via $w \mapsto F(\mu)^{-1}w$. These identities relate adjoints in dual coordinates. The vector $F\nabla_\mu L$ is also the natural-gradient direction in mean coordinates, whose metric is F^{-1}. For a fixed-message log normalizer it is the posterior moment correction in Eq. (30).

To see this in action, take the mean-form ELBO $L(\mu;\theta) = \langle \theta, \mu\rangle - A(\theta) - A^*(\mu)$. A forward pass might: (i) from η (or θ) compute $\mu = \nabla A(\eta)$; (ii) evaluate L using μ and $A^*(\mu)$. The reverse pass then starts with $\partial L/\partial \mu = \theta - \nabla A^*(\mu) = \theta - \eta$ and sends it back to η by multiplying with $F(\eta)$, giving $\partial L/\partial \eta = F(\eta)(\theta - \eta)$. Setting this to zero recovers $\theta = \eta$ — the usual dual consistency $\mu = \nabla A(\eta)$ and $\eta = \nabla A^*(\mu)$ appearing as first-order optimality conditions. In other words, the same convex-conjugacy equations that define the family also show up as the stationary point where the reverse-mode pullback vanishes.

On standard sites the formulas reduce to familiar backprop patterns. For a categorical site, $\mu = \mathrm{softmax}(\eta)$ comes from $A(\eta) = \log \sum_j e^{\eta_j}$; its VJP is the "softmax-covariance" map $(\mathrm{diag}(\mu) - \mu\mu^\top)v$, whose output is a zero-sum direction in the simplex tangent. For a Gaussian with both mean and covariance free, the Fisher acts on the sufficient statistics $(z, zz^\top)$; it is not simply the covariance of z. The Gaussian coordinate differentials above and the full signal conversion in Appendix B.5 give the corresponding maps.

At the graph level, reverse differentiation of exact message computations, including their normalizers, gives the adjoints of log evidence. At a proper normalized cut reference, local Fisher conversion gives the posterior moment correction in Eq. (30). On a discrete chain this recovers the forward–backward smoothing correction; on a linear–Gaussian chain it recovers the RTS correction. The examples below specify the normalized interfaces and future likelihoods that are differentiated.

Two practical consequences follow. Module adjoints can be reused across inference objectives, with the belief family supplying their Fisher conversion. The objective and finite update then specify the inference method: a fixed-message log normalizer gives an EP moment correction, whereas a variational objective uses its own canonical mirror update. Damping and valid-domain conditions remain part of that finite update.

Schur complement and constrained natural gradients on trees.

The Schur complement provides a structured linear-algebra description of the negative-ELBO natural-gradient direction. This variational calculation is distinct from differentiating the local cut log evidence in Eq. (30).

We begin with a constrained quadratic problem central to the concept of natural-gradient steps in Bayesian inference. Let $\mu = \mu_C, \mu_S$ represent stacked means from cliques and separators of a graphical model. Define the gradient of our objective function $\Phi(\mu)$ at the current point

as $g = \nabla_\mu \Phi(\mu)$. Additionally, define the block-diagonal entropy Hessian in mean coordinates as: $H = \operatorname{diag}\big(\{F_C^{-1}\}_{C\in\mathcal{C}}, \{-F_S^{-1}\}_{S\in\mathcal{S}}\big)$, where F_C and F_S are local Fisher information blocks associated with cliques and separators respectively.

As established in Appendix A, H is positive definite on the consistency tangent for the full compatible tree families, although it is indefinite in unrestricted region coordinates. Let B be the constraint Jacobian, with a row $[-M_{C\to S} \mid I]$ for each clique–separator incidence. The direction solves

$$\min_d \langle g, d\rangle + \tfrac{1}{2} d^\top H d, \qquad Bd = 0,$$

with KKT equations

$$Hd + g + B^\top \lambda = 0, \qquad Bd = 0.$$

There is one multiplier λ_{CS} per incidence. The block equations are

$$d_C = -F_C \left(g_C - \sum_{S\subset C} M_{C\to S}^\top \lambda_{CS} \right), \qquad d_S = F_S \left(g_S + \sum_{C\supset S} \lambda_{CS} \right),$$

where $g_C = \eta_C - \theta_C$ and $g_S = -\eta_S - \theta_S$. With invertible region Fisher blocks and redundant constraint rows removed, eliminating d gives

$$(BH^{-1}B^\top)\lambda = -BH^{-1}g, \qquad d = -H^{-1}(g + B^\top \lambda).$$

The negative sign follows directly by substituting the second equation into $Bd = 0$. Equivalently, a basis R for $\ker B$ gives the positive-definite reduced system

$$(R^\top H R)\delta = -R^\top g, \qquad d = R\delta.$$

These equations describe the local constrained direction. They do not identify a separator multiplier with a cut-evidence adjoint.

The associated entropy-mirror path is defined in Eq. (27). Differentiating its stationarity conditions at zero step size yields precisely this KKT system, while a unit mirror minimization reaches the exact posterior endpoint. Thus the negative-ELBO natural gradient is the initial tangent of a mirror path. The cut-evidence natural gradient in Eq. (30) is instead already a finite posterior moment correction at its normalized reference.

Tree structure is also useful for solving Gaussian information systems: block elimination on a chain leads to the familiar Kalman/RTS recursions. The examples below connect those recursions to reverse differentiation of the specified evidence computation. This keeps the linear-algebra and autodiff descriptions tied to their respective objectives and coordinates.

Examples.

Discrete HMM (reverse–mode backprop of log evidence in our notation). Let $z_t \in \{1, \dots, K\}$ be hidden states and y_t observations. Denote the transition matrix by $P \in \mathbf{R}^{K\times K}$ with $P_{ij} = \mathbf{P}(z_t = j \mid z_{t-1} = i)$ and the emission likelihood vector at time t by $e_t \in \mathbf{R}^K$ with $e_t(j) = \mathbf{P}(y_t \mid z_t = j)$. The normalized forward (filtering) mean at time t is $\alpha_t \in \Delta^K$ with recursion

$$m_t = P^\top \alpha_{t-1}, \qquad \tilde{\alpha}_t = e_t \odot m_t, \qquad s_t = \mathbf{1}^\top \tilde{\alpha}_t, \qquad \alpha_t = \tilde{\alpha}_t / s_t,$$

and we view the separator (node) canonical parameters as the logits $\eta_t := \log \alpha_t - \frac{1}{K}\mathbf{1}\,\mathbf{1}^\top \log \alpha_t$ (log–probabilities projected to the simplex tangent), so that $\alpha_t = \nabla A(\eta_t)$ with $A(\eta) = \log \sum_j e^{\eta_j}$.

We consider the log evidence $L := \log p(y_{1:T})$. At the normalized filter, hold the past and model fixed and define the future likelihood $\beta_t(j) = p(y_{t+1:T} \mid z_t = j)$. The downstream scalar is $\mathcal{L}_t(\eta) = \log \sum_j \operatorname{softmax}(\eta)_j \beta_t(j)$. Below, $\nabla_{\eta_t} L$ denotes this internal interface adjoint at the filtering logits, with positive filtering probabilities and nonzero normalizers. Two identities fix the base case. First, for the categorical site,

$$\nabla_{\eta_t} \log \alpha(z_t; \eta_t) = x_t - \alpha_t, \qquad \text{so} \quad \nabla_{\eta_t} L \;=\; \gamma_t - \alpha_t,$$

where x_t is the one–hot vector for z_t, and $\gamma_t \in \Delta^K$ are the exact smoothing marginals $\gamma_t(j) = \mathbf{P}(z_t = j \mid y_{1:T})$. Second, the Fisher block on the simplex tangent is $F(\eta_t) = \nabla^2 A(\eta_t) =$

$\mathrm{diag}(\alpha_t) - \alpha_t \alpha_t^\top$ so that mirror VJPs multiply covectors by $F(\eta_t)$ and mirror JVPs multiply tangents by $F(\eta_t)$.

The only nontrivial step is the Jacobian of the map $\eta_{t-1} \mapsto \eta_t$ induced by the normalized forward update. A convenient form is the differential identity for the *log*–forward marginals:

$$\frac{\partial \log \alpha_t(j)}{\partial \eta_{t-1}(i)} \;=\; \underbrace{\mathbf{P}(z_{t-1} = i \mid z_t = j,\, y_{1:t-1})}_{\text{backward kernel } K_{t-1|t}(i|j)} \;-\; \underbrace{\mathbf{P}(z_{t-1} = i \mid y_{1:t})}_{\text{one–step smoothing } \gamma_{t-1|t}(i)} \quad .$$

To verify, write $\alpha_t(j) = \frac{e_t(j) \sum_i \alpha_{t-1}(i) P_{ij}}{\sum_{j'} e_t(j') \sum_i \alpha_{t-1}(i) P_{ij'}}$ and differentiate with respect to $\eta_{t-1}(i)$, using $\partial \alpha_{t-1}(k)/\partial \eta_{t-1}(i) = \alpha_{t-1}(k)(\delta_{ki} - \alpha_{t-1}(i))$ and the quotient rule. This Jacobian depends only on forward quantities: $K_{t-1|t}(i \mid j) = \frac{\alpha_{t-1}(i) P_{ij}}{m_t(j)}$ with $m_t = P^\top \alpha_{t-1}$, and $\gamma_{t-1|t}(i) = \frac{\alpha_{t-1}(i) \sum_j P_{ij} e_t(j)}{s_t} = \mathbf{P}(z_{t-1} = i \mid y_{1:t})$.

With this Jacobian, reverse–mode through one HMM step is now a one–line VJP. Let $g_t := \nabla_{\eta_t} L$ denote the separator–canonical adjoint at time t (a zero–sum vector on the simplex tangent). The chain rule gives

$$\nabla_{\eta_{t-1}(i)} L \;=\; \sum_j g_t(j)\, \frac{\partial \log \alpha_t(j)}{\partial \eta_{t-1}(i)} \;+\; \frac{\partial \log p(y_t \mid y_{1:t-1})}{\partial \eta_{t-1}(i)}.$$

The second term is the derivative of the per–step log normalizer: $\log p(y_t \mid y_{1:t-1}) = \log s_t$ with $s_t = \sum_j e_t(j) m_t(j)$ and $m_t = P^\top \alpha_{t-1}$, yielding

$$\frac{\partial \log p(y_t \mid y_{1:t-1})}{\partial \eta_{t-1}(i)} \;=\; \mathbf{P}(z_{t-1} = i \mid y_{1:t}) \;-\; \alpha_{t-1}(i) \;=\; \gamma_{t-1|t}(i) - \alpha_{t-1}(i).$$

Collecting terms and writing in vector form,

$$\nabla_{\eta_{t-1}} L \;=\; K_{t-1|t}\, g_t \;+\; \big(\gamma_{t-1|t} - \alpha_{t-1}\big), \qquad K_{t-1|t}(i \mid j) = \frac{\alpha_{t-1}(i) P_{ij}}{(P^\top \alpha_{t-1})(j)}.$$

This recovers the cut-evidence correction in the HMM example of Appendix A; centering the output logits leaves this VJP unchanged because g_t has zero sum. Two properties make the recursion numerically and conceptually clean. First, $K_{t-1|t}$ is column–stochastic (each column sums to one), so it preserves the simplex tangent: if $\mathbf{1}^\top g_t = 0$ then $\mathbf{1}^\top K_{t-1|t} g_t = 0$. Second, initializing at $t = T$ with $g_T = \gamma_T - \alpha_T = 0$ and running the recursion backward yields $g_t = \gamma_t - \alpha_t$ for all t; this follows by a one–line substitution:

$$g_{t-1} \;=\; K_{t-1|t}(\gamma_t - \alpha_t) + (\gamma_{t-1|t} - \alpha_{t-1}) \;=\; \gamma_{t-1} - \alpha_{t-1},$$

because $\gamma_{t-1} = \sum_j \mathbf{P}(z_{t-1} \mid z_t = j, y_{1:t-1})\, \gamma_t(j)$ and $\alpha_t = \mathbf{P}(z_t \mid y_{1:t})$ imply $K_{t-1|t} \alpha_t = \gamma_{t-1|t}$. Thus the reverse–mode computation reproduces the analytical gradient.

For completeness, if gradients are needed in mean space rather than logits, multiply by the inverse Fisher on the simplex tangent: $\nabla_{\alpha_t} L = F(\alpha_t)^{-1} \nabla_{\eta_t} L$, where $F(\alpha_t) = \mathrm{diag}(\alpha_t) - \alpha_t \alpha_t^\top$. In practice, operating in canonical space (logits) avoids explicit pseudo–inversion: the recursion above already stays in the tangent and is numerically stable under standard forward scaling (e.g., with $\log s_t$).

Linear–Gaussian chain (reverse–mode backprop of log evidence in our notation). Let $x_t \in \mathbf{R}^d$ evolve as $x_t = F_t x_{t-1} + \omega_t$, $\omega_t \sim \mathcal{N}(0, Q_t)$, and let $y_t = H_t x_t + \nu_t$, $\nu_t \sim \mathcal{N}(0, R_t)$. Assume independent noises, $Q_t, R_t \succ 0$, and a nonsingular initial Gaussian. The Kalman *filter* provides the node (separator) mean and covariance

$$\begin{aligned} m_{t|t-1} &= F_t\, m_{t-1|t-1}, \\ P_{t|t-1} &= F_t P_{t-1|t-1} F_t^\top + Q_t, \\ S_t = H_t P_{t|t-1} H_t^\top + R_t, &\qquad K_t = P_{t|t-1} H_t^\top S_t^{-1}. \end{aligned}$$

$$m_{t|t} = m_{t|t-1} + K_t\, \varepsilon_t, \qquad P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \qquad \varepsilon_t = y_t - H_t m_{t|t-1}.$$

We view the separator exponential family as Gaussian with mean parameters $\mu_{S_t} = (m_{t|t}, M_{t|t})$, where $M_{t|t} = \mathbf{E}[x_t x_t^\top] = P_{t|t} + m_{t|t} m_{t|t}^\top$, and canonical parameters $\eta_{S_t} = (\eta_{1,t}, \eta_{2,t})$ with $\eta_{1,t} = P_{t|t}^{-1} m_{t|t}$, $\eta_{2,t} = -\frac{1}{2} P_{t|t}^{-1}$. The per–step log evidence is $\ell_t = \log \mathcal{N}(y_t;\, H_t m_{t|t-1},\, S_t)$ and the total evidence is $L = \sum_t \ell_t$. As in the HMM, the internal adjoints at a normalized filtering state differentiate the fixed future likelihood: $\mathcal{L}_t(m, P) = \log \int p(y_{t+1:T} \mid x)\, \mathcal{N}(x; m, P)\, dx$. These derivatives use stored mean/covariance coordinates (m, P), distinct from the exponential-family mean coordinates (m, M).

Seed gradients (likelihood) at time t with respect to the predicted mean and covariance are

$$\frac{\partial \ell_t}{\partial m_{t|t-1}} \;=\; H_t^\top S_t^{-1}\, \varepsilon_t, \qquad \frac{\partial \ell_t}{\partial P_{t|t-1}} \;=\; \tfrac{1}{2}\, H_t^\top \big(S_t^{-1} \varepsilon_t \varepsilon_t^\top S_t^{-1} - S_t^{-1}\big) H_t,$$

which are the standard Gaussian VJPs for mean and covariance of a linear transform.

VJPs through the Kalman update $(m_{t|t-1}, P_{t|t-1}) \mapsto (m_{t|t}, P_{t|t})$ can be written without omitted gain terms. Put $U_t = I - K_t H_t$ and $r_t = H_t^\top S_t^{-1} \varepsilon_t$. For fixed H_t, R_t, y_t, the complete differentials are

$$\begin{aligned} dm_{t|t} &= U_t\, dm_{t|t-1} + U_t (dP_{t|t-1}) r_t, \\ dP_{t|t} &= U_t (dP_{t|t-1}) U_t^\top. \end{aligned}$$

Indeed, $dK_t = U_t (dP_{t|t-1}) H_t^\top S_t^{-1}$ supplies the covariance contribution to the mean differential. The covariance differential already includes the gain dependence. For symmetric covariance adjoints and $\mathrm{sym}(X) = (X + X^\top)/2$, the update VJP is

$$\bar{m}_{t|t-1} \mathrel{+}= U_t^\top g_{m_t}, \qquad \bar{P}_{t|t-1} \mathrel{+}= U_t^\top G_{P_t} U_t + \mathrm{sym}(U_t^\top g_{m_t} r_t^\top).$$

The symmetric cross term follows by pairing $g_{m_t}^\top U_t (dP_{t|t-1}) r_t$ with the symmetric covariance perturbation under the trace inner product.

Reverse recursion in mean/covariance coordinates. Let $g_{m_t} = \partial_m \mathcal{L}_t$ and $G_{P_t} = \partial_P \mathcal{L}_t = G_{P_t}^\top$ at the filtering state. Combining the update VJP, the likelihood seeds above, and the prediction map gives

$$\begin{aligned} g_{m_{t-1}} &= F_t^\top (U_t^\top g_{m_t} + r_t), \\ G_{P_{t-1}} &= F_t^\top \Big[U_t^\top G_{P_t} U_t + \mathrm{sym}(U_t^\top g_{m_t} r_t^\top) \\ &\qquad + \tfrac{1}{2} \big(r_t r_t^\top - H_t^\top S_t^{-1} H_t \big) \Big] F_t. \end{aligned}$$

The terminal future likelihood is constant, so $g_{m_T} = 0$ and $G_{P_T} = 0$. Each innovation likelihood is added once at its predicted state during this recursion. The identity $P_{t|t} P_{t|t-1}^{-1} = U_t$ relates this expression to the usual predicted-to-filtered information map.

Canonical vs. mean gradients and the RTS identity. In the Gaussian exponential family, the separator–canonical gradient of the fixed future log evidence $\mathcal{L}_t$ has the *moment–difference* form

$$\nabla_{\eta_{1,t}} \mathcal{L}_t \;=\; m_{t|T} - m_{t|t}, \qquad \nabla_{\eta_{2,t}} \mathcal{L}_t \;=\; M_{t|T} - M_{t|t},$$

where $(\cdot)_{|T}$ denote RTS–smoothed moments. Mean–space gradients are obtained by the mirror VJP $g_\mu = F^{-1} g_\eta$. Writing $M_{t|T} = P_{t|T} + m_{t|T} m_{t|T}^\top$ and $M_{t|t} = P_{t|t} + m_{t|t} m_{t|t}^\top$, conversion from raw moments to the stored (m, P) coordinates gives

$$\frac{\partial \mathcal{L}_t}{\partial m_{t|t}} \;=\; P_{t|t}^{-1}\, (m_{t|T} - m_{t|t}), \qquad \frac{\partial \mathcal{L}_t}{\partial P_{t|t}} \;=\; \tfrac{1}{2} \Big(P_{t|t}^{-1} P_{t|T} P_{t|t}^{-1} \;-\; P_{t|t}^{-1} + P_{t|t}^{-1} \Delta_t \Delta_t^\top P_{t|t}^{-1} \Big),$$

with $\Delta_t := m_{t|T} - m_{t|t}$. These equalities summarize what the reverse recursion computes: the reverse–mode adjoints coincide with the information–form RTS backward quantities. The positive outer-product term follows from $\mathbb{E}[(x_t - m_{t|t})(x_t - m_{t|t})^\top \mid y_{1:T}] = P_{t|T} + \Delta_t \Delta_t^\top$ in the Gaussian

covariance score. Encoding these adjoints gives the exact first- and raw-second-moment corrections, while the Kalman–RTS example in Appendix A computes the same moments by Gaussian conditioning.

Remarks. (i) The recursion preserves symmetry of G_{P_t} and uses the usual Kalman linear solves. (ii) Canonical adjoints can be carried by applying the full Gaussian Fisher conversions at each interface; the underlying VJP dependencies must still be retained. (iii) The local cut-evidence identity explains why the converted adjoints give RTS moment corrections. The separate negative-ELBO mirror endpoint and its initial natural-gradient tangent are described in Appendix A.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the single-pass uncertainty contribution. Sections 2–3 distinguish exact tree results from projected neural updates.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 3 discusses posterior factorization, nonlinear closure, shared-variable covariance, and stochastic-update scope, and connects these approximations to computation cost.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorems 3.1–3.2 give the main statements and proof sketches. Appendices A–B provide the exact-tree and projected-EP derivations; Appendix B also gives the covariance extensions, and Appendix C gives the MFVI derivations.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The algorithm, observation model, datasets, and protocol families are described in Sections 3–4 and Appendix D. The paper does not enumerate every run-specific setting for the original benchmark tables.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The work uses publicly available benchmark datasets and describes implementation details in Appendix D; a public release reproducing every main-table result is not linked in this paper.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: Appendix D provides the main experimental setup and the separate additional-study protocols. Some original run-specific prior, damping, and schedule settings are not enumerated.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The main tables report means over three training seeds without seed-level confidence intervals or statistical significance tests.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Table 4 gives relative WRN prediction costs and a training-step ratio. A complete hardware specification and per-run compute inventory are not reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses established benchmark datasets and does not collect new human-subject data. The Broader impacts section discusses beneficial uses and risks of predictive uncertainty.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Broader impacts section discusses selective prediction and escalation, together with risks from miscalibration, distribution shift, and misuse in high-stakes applications.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk generative model or a newly collected sensitive dataset.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Dataset and model creators are cited, but the paper does not include a complete explicit inventory of asset versions, licenses, and terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The appendices describe module calculations, derivations, and experimental protocols. A complete public code artifact covering all original benchmarks is not specified.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not conduct crowdsourcing or recruit human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not conduct new human-subject experiments requiring institutional review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Language models serve as evaluated predictive backbones; an LLM agent is not part of the proposed inference or learning algorithm.