DIRECT MESSAGE APPROXIMATION (DMA): A CONSISTENCY-BASED FRAMEWORK FOR TRACTABLE APPROXIMATE INFERENCE ON FACTOR GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Approximate message passing on factor graphs underlies two dominant families of probabilistic inference algorithms: expectation propagation (EP) and variational message passing (VMP). Both methods approximate the *marginal* at each factor edge, forcing an iterative round-robin schedule, risking negative-precision messages, and, for VMP, collapsing to point estimates at Dirac-delta factors. We introduce *Direct Message Approximation* (DMA), which approximates factor-to-variable *messages* directly rather than the marginal. For normalisable factors, we define a consistency condition (requiring exactness when all other incoming messages are Dirac deltas) to guide message construction. We prove a master theorem (proper messages, any graph) bounding marginal KL from message KL, with three structural corollaries: Dirac-input consistency, no EP-style inner-loop iteration, and no negative-precision messages. Further, we prove a complementary $O(1/r^2)$ guarantee for the inherently improper backward message of the product factor, whose closed-form treatment has resisted prior work. As a concrete instantiation, we derive explicit DMA messages for the product and leaky-ReLU factors and assemble a Bayesian neural network (BNN) inference algorithm with one forward/backward sweep per training example and no gradient learning-rate hyperparameter, validating that the structural guarantees translate to predictive uncertainty that widens in data-sparse regions, including under model mismatch.

1 INTRODUCTION

Factor graphs provide a unifying language for probabilistic inference: a joint density factorises into local potentials, and the sum-product algorithm computes exact marginals on trees by passing messages along edges (Kschischang et al., 2001). When factors are non-conjugate to the message family (as in neural network likelihoods, latent Dirichlet models, and models with nonlinear potentials), the factor-to-variable integral is intractable, and approximate message passing is required.

Two classical strategies dominate. Expectation propagation (EP; Minka, 2001) minimises the forward KL divergence from the true posterior marginal to a tractable exponential-family approximation, then recovers the approximate factor-to-variable message by dividing out the variable-to-factor message at the same edge. Variational message passing (VMP; Winn & Bishop, 2005) instead maximises the evidence lower bound under a factorised approximation, minimising the reverse KL. Despite their theoretical differences, both methods share a structural commitment: they approximate the *marginal* $\hat{p}_{X_j}$ at each factor edge, and recover the outgoing factor-to-variable message by

$$\hat{m}_{f\to X_j}(\cdot) \;=\; \hat{p}_{X_j}(\cdot) \,/\, m_{X_j\to f}(\cdot). \tag{1}$$

Three pathologies of marginal-based approximation. This marginal-first design induces three operational problems, each well-known in the EP and VMP literature.

1. **Iterative schedule.** Because $\hat{m}_{f\to X_j}$ depends on the current incoming message $m_{X_j\to f}$ via the division in (1), updating one outgoing message changes the variable-to-factor message at the same edge, invalidating the outgoing messages of all neighbouring factors. The only remedy is a round-robin sweep that iterates until a fixed point is reached, typically requiring many passes over the entire factor graph (Minka, 2001).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2. **Invalid messages in Gaussian EP.** When the projected Gaussian marginal $\hat{p}_{X_j}$ is wider than the incoming message $m_{X_j \to f}$, the division in (1) produces a Gaussian with negative precision. Such messages are invalid probability distributions; they arise routinely when a factor disperses mass (e.g. a product of two uncertain variables) and can destabilise subsequent updates by propagating indefinitely.
3. **Dirac-delta collapse in VMP.** For a factor $f(\mathbf{x}) = \delta(x_j - g(\mathbf{x}_{-j}))$ encoding a deterministic relation, the VMP update (1) collapses $\hat{p}_{X_j}$ to a Dirac delta, setting $\hat{m}_{f \to X_j}$ to a point mass and eliminating all posterior variance. Since every parameterised layer in a feedforward neural network is such a Dirac-delta factor, VMP cannot maintain a non-trivial weight posterior in Bayesian neural networks (Winn & Bishop, 2005).

Our proposal. We argue that the root cause of all three pathologies is the division in (1): it creates an incoming-message dependence that forces iteration and can produce invalid outputs. DMA eliminates cavity division from the construction of approximate factor-to-variable messages by approximating the *message* $\hat{m}_{f \to X_j}$ directly rather than the marginal.

The natural design criterion for a message approximation is a *consistency condition*: it should agree with the exact sum-product message whenever all other inputs to the factor are point masses. In that limit the factor-to-variable integral reduces to a deterministic function evaluation and is always tractable, so the exact message is known and we can demand recovery. We formalise this as Definition 2.1 and call the resulting approximation a *direct message approximation* (DMA). Because no per-factor message computation requires the ratio (1), the iterative schedule is unnecessary, negative-precision messages cannot arise from message construction itself, and Dirac-delta factors satisfy the consistency condition exactly in the concentrated-input limit. (Training does use a division when replacing old factor messages across epochs, see Algorithm 1, but this outer-loop replacement is distinct from the per-factor computation and does not create the feedback loop or negative-precision pathologies of (1).)

The key distinction from assumed-density filtering (ADF; Lauritzen, 1992; Opper & Winther, 1999), which also skips cavity division, is that DMA adds a formal *consistency axiom* that defines the admissible direct-approximation principle, here instantiated through moment-matching projection onto the Gaussian family (Section 2.2), an *edge-local* master theorem bounding marginal KL from message KL without any graph contraction condition (Section 2), and an $\mathrm{O}(1/r^2)$ guarantee for the improper product-backward message that falls outside the classical moment-matching scope (Section 3); see Section 5 for a detailed comparison with EP and ADF.

Contributions.

1. **DMA framework and master theorem (Section 2).** We define a *consistency condition* as the design criterion for direct message approximation, give a construction recipe via moment matching, and prove a *master theorem* (Theorem 2.3) bounding the marginal KL at any edge by $O(\delta)$ in the message KL δ, edge-locally and without any graph contraction condition. The closest prior result (Ihler et al., 2005) bounds perturbations in loopy BP via graph-level contraction rates; our bound applies to EP-style projected messages and holds on any graph.
2. **Product-factor DMA construction (Section 3).** We derive DMA messages for the product factor $\delta(z - xy)$, the central and technically most demanding case: it is the main factor required for BNN weight updates, where the backward distribution of $X = Z/Y$ has no closed-form Gaussian representation and where the standard EP backward message is structurally improper. We use a *log-normal intermediate technique* to characterise the ratio distribution. Moreover, we prove an $O(1/r^2)$ accuracy bound in the concentrated-input regime (Theorem 3.3), an analysis outside the scope of the master theorem. The complementary activation factor $\delta(y - \mathrm{ReLU}_\alpha(x))$ is handled by truncated-Gaussian moment matching; both factors verify the consistency condition analytically.
3. **Instantiation and validation (Section 4).** The two factors are sufficient to assemble a complete BNN inference algorithm. Three corollaries of Theorem 2.3 give: no EP-style inner-loop iteration (Corollary 2.5), no negative-precision messages by construction (Corollary 2.6), and Dirac-input consistency on all linear factors (Corollary 2.4). Exper-

iments confirm convergence, positive weight variances, and structurally widening extrapolation uncertainty; a 23× larger network (Appendix G) confirms the procedure remains computationally viable without algorithmic changes.

Sections 2–5 cover the DMA framework, factor derivations, the BNN algorithm, and related work; background, all proofs and experimental details are in the Appendix A - G.

2 Direct Message Approximation

Notation. We follow the factor-graph and message-passing conventions of Appendix A; readers unfamiliar with the sum-product algorithm may consult it for background. We write $m_{f\to X_j}$ for the factor-to-variable message on edge (f, X_j) and $m_{X_j\to f}$ for the variable-to-factor message. The marginal at X_j is $p_{X_j} = \prod_{i\in \mathrm{ne}(X_j)} m_{f_i\to X_j}$, and $\mathcal{Q}$ denotes a fixed exponential family (Gaussian throughout this paper). $\mathrm{KL}[\cdot \,\|\, \cdot]$ is the forward KL divergence.

DMA approximates the outgoing message directly, without forming the marginal at all; this avoids the division in (1) and all three pathologies it creates. This section formalises the approach, proves a master theorem bounding the resulting marginal error, and derives three structural corollaries for BNN inference.

2.1 Definition and Consistency Condition

The exact factor-to-variable message (16) is, in general, intractable. There is, however, one regime in which it is always tractable: when all other incoming messages are Dirac deltas $\delta(\cdot - x_k)$. Then the multi-dimensional integral collapses to a single factor evaluation,

$$m_{f_i\to X_j}(x_j) = \int f_i(x_j, \mathbf{x}_{-j}) \prod_{k\neq j} \delta(x_k - \bar{x}_k)\, \mathrm{d}\mathbf{x}_{-j} = f_i(x_j, \bar{\mathbf{x}}_{-j}), \tag{2}$$

which is just the factor evaluated at the fixed point values $\bar{\mathbf{x}}_{-j}$. The resulting message is an unnormalised one-dimensional density in x_j; for Gaussian $\mathcal{Q}$ and the common case where $f_i(\cdot, \bar{\mathbf{x}}_{-j})$ is a Gaussian likelihood or a pushforward of a Gaussian through a smooth map, the projection onto $\mathcal{Q}$ is tractable in closed form.

The consistency condition asks that a message approximation converge to the exact message as inputs concentrate toward this tractable limit.

Definition 2.1 (Direct Message Approximation)**.** *Let f_i be a factor in a factor graph with neighbours $\mathrm{ne}(f_i)$, let $X_j \in \mathrm{ne}(f_i)$, and suppose $f_i(\cdot, \bar{\mathbf{x}}_{-j})$ is normalisable for every $\bar{\mathbf{x}}_{-j}$. A family of distributions $\hat{m}_{f_i\to X_j}$, parametrised by the incoming messages $\{m_{X_k\to f_i}\}_{k\neq j}$, from an exponential family $\mathcal{Q}$ is a* direct message approximation *(DMA) if it satisfies the* concentration consistency condition*: for every $\bar{\mathbf{x}}_{-j} \in \mathbb{R}^{|\mathrm{ne}(f_i)|-1}$, if $m_{X_k\to f_i} = \mathcal{N}(\bar{x}_k, \sigma_k^2)$ for all $k \neq j$, then as $\sigma_k \to 0$,*

$$\hat{m}_{f_i\to X_j} \xrightarrow{w} \frac{f_i(\cdot,\ \bar{\mathbf{x}}_{-j})}{\displaystyle\int f_i(x_j,\ \bar{\mathbf{x}}_{-j})\, \mathrm{d}x_j}, \tag{3}$$

where $\xrightarrow{w}$ denotes weak convergence of probability measures and the right-hand side is the normalised exact message (2) evaluated at the concentrated inputs.

Remark 2.2. *The normalisability condition in Definition 2.1 holds for all factor-edge pairs in this paper except one: the product factor* backward *message is improper for general Gaussian inputs (Remark B.2). This case is handled by a separate log-normal intermediate construction whose approximation quality is characterised by Theorem 3.3: the KL to a truncated proper reference is $O(1/r^2)$ in the input signal-to-noise ratio r, recovering concentration consistency in the limit $r \to \infty$.*

The consistency condition is a necessary but not sufficient design criterion: the moment-matching construction (Section 2.2) selects a specific DMA satisfying it, and Theorem 2.3 bounds the marginal error at any input width—consistency, construction, and master theorem together constitute the local theoretical guarantees for individual DMA message computations.

2.2 Construction Recipe

For the class of factors that arises in Bayesian neural networks — Dirac-delta factors of the form $f(\mathbf{x}) = \delta(x_j - g(\mathbf{x}_{-j}))$ encoding a continuous deterministic function g — there is a systematic recipe for constructing a Gaussian DMA.

Recipe. Let the incoming messages $\{m_{X_k \to f}\}_{k \neq j}$ be Gaussians $\mathcal{N}(\bar{x}_k, \sigma_k^2)$.

1. **Compute moments.** Treat the inputs $\{X_k\}_{k \neq j}$ as independent Gaussian random variables with the given means and variances, and compute the first two moments of $x_j = g(\mathbf{x}_{-j})$ under this joint:
$$\mu_{x_j} := \mathbb{E}[g(\mathbf{X}_{-j})], \qquad \sigma^2_{x_j} := \mathrm{Var}[g(\mathbf{X}_{-j})].$$
2. **Project.** Set $\hat{m}_{f \to X_j} := \mathcal{N}(\mu_{x_j}, \sigma^2_{x_j})$ via Theorem A.1 (moment matching minimises the forward KL to the Gaussian family).

Consistency verification. In the Dirac limit $\sigma_k \to 0$ for all $k \neq j$, the inputs concentrate on their means $\bar{x}_k$, so $\mu_{x_j} \to g(\bar{\mathbf{x}}_{-j})$ and $\sigma^2_{x_j} \to 0$. The approximate message converges to $\delta(\cdot - g(\bar{\mathbf{x}}_{-j}))$, which equals the exact message (2) (a point mass at the function value). The KL divergence between two identical distributions is zero, so the consistency condition (3) holds.

2.3 Master Theorem and Consequences

The master theorem bounds the error in the marginal p_{X_j} that results from using a DMA in place of the exact message. The bound is edge-local: it depends only on the quality δ of the single outgoing message, and on the sup-norm of the incoming message at the same edge.

Theorem 2.3 (DMA Master Theorem). *Let f be a factor with target variable X_j. Let $m_{f \to X_j}$ be the true normalised sum-product message and $\hat{m}_{f \to X_j}$ a DMA; write $\delta := \mathrm{KL}\big[m_{f \to X_j} \,\|\, \hat{m}_{f \to X_j}\big] < \infty$. Let $m_{X_j \to f}$ be a normalised incoming message at the same edge, define $p_{X_j} := m_{f \to X_j} \cdot m_{X_j \to f}$ and $\hat{p}_{X_j} := \hat{m}_{f \to X_j} \cdot m_{X_j \to f}$, with normalising constants Z and $\hat{Z}$ respectively. Then*

$$\mathrm{KL}\left[\frac{p_{X_j}}{Z} \,\|\, \frac{\hat{p}_{X_j}}{\hat{Z}}\right] \;\leq\; \frac{\|m_{X_j \to f}\|_\infty}{Z}\,\delta. \tag{4}$$

Proof. The full proof is in Appendix B.2.

Interpretation. The bound is linear in δ; for $\delta < 1$ (the approximation regime) this is strictly tighter than a square-root dependence. The bound is *edge-local* and requires no global contraction condition, contrasting with Ihler et al. (2005) whose analogous result for loopy BP uses graph-level contraction rates. Across 192 leaky-ReLU factor configurations spanning a range of slopes, input widths, and SNR values (Appendix C.1), the bound holds in every case with a maximum normalised ratio of 0.97, directly certifying the theorem.

The corollaries below extract the structural consequences of the master theorem for BNN inference.

Corollary 2.4 (Asymptotic Dirac-Input Consistency). *Let $f(\mathbf{x}) = \delta(x_j - g(\mathbf{x}_{-j}))$, suppose the DMA is constructed by the recipe of Section 2.2, and assume g is twice continuously differentiable in a neighbourhood of $\bar{\mathbf{x}}_{-j}$ with $\nabla g(\bar{\mathbf{x}}_{-j}) \neq \mathbf{0}$, and regular in the tails. Then, with each incoming message $m_{X_k \to f} = \mathcal{N}(\bar{x}_k, \sigma_k^2)$,*

$$\lim_{\sigma_k \to 0} \delta \;=\; \lim_{\sigma_k \to 0} \mathrm{KL}\big[m_{f \to X_j} \,\|\, \hat{m}_{f \to X_j}\big] \;=\; 0.$$

The result extends to piecewise-C^2 factors (e.g. leaky-ReLU) by treating isolated non-smooth points separately; the upper bound in (4) therefore vanishes in this limit.

Corollary 2.5 (No EP Inner-Loop Iteration). *Because $\hat{m}_{f \to X_j}$ does not depend on $m_{X_j \to f}$ (cf. Definition 2.1), all factor-to-variable messages for a single training example can be computed in one forward sweep followed by one backward sweep, with no EP-style inner-loop fixed-point iteration between individual message computations. This is distinct from the outer training loop, which repeats sweeps over examples and epochs until the weight beliefs converge.*

Corollary 2.6 (No Negative-Precision Messages)**.** *Because DMA produces a valid member of $\mathcal{Q}$ by construction (via Theorem A.1), no approximate message can have negative precision. The pathology arises in EP when the projected marginal is wider than the incoming message; DMA never forms this ratio.*

Proofs of all three corollaries are in Appendix B.3. Section 3 derives explicit DMA messages for the product and ReLU factors, the two non-conjugate building blocks needed for BNNs. Section 4 then assembles these into a complete inference algorithm with one forward/backward factor sweep per training example and no EP-style inner-loop iteration.

3 DMA FOR NON-GAUSSIAN FACTORS

Two non-conjugate factors appear in every feedforward BNN and cannot be handled by the exact sum-product algorithm: the *product factor* $\delta(z - xy)$ (elementwise weight-activation products) and the *ReLU factor* $\delta(y - \mathrm{ReLU}_\alpha(x))$ (activation nonlinearity). The Gaussian prior $\mathcal{N}(w;\, 0, \sigma_0^2)$ and likelihood $\mathcal{N}(y;\, z, \beta^2)$ are conjugate to the message family, so their messages are exact and $\delta = 0$ at both edges (Winn & Bishop, 2005). We apply the construction recipe of Section 2.2 to the two non-conjugate factors, yielding closed-form Gaussian DMA messages. The product factor has exact closed-form forward moments (via independence of X and Y), so the only approximation is the Gaussian projection; its backward message is inherently improper. The ReLU factor has a proper backward message for all $\alpha > 0$, with an explicit normalisation correction that depends on α.

3.1 THE PRODUCT FACTOR

Factor definition. The product factor encodes the deterministic relation $z = xy$:

$$f(x, y, z) = \delta(z - xy)\,. \tag{5}$$

For Gaussian inputs $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, the product $Z = XY$ is not Gaussian, so the exact forward message to Z is not in $\mathcal{Q}$. We apply the recipe: compute the exact moments of Z under the independent Gaussian inputs, then project onto $\mathcal{N}$.

Forward message. The moments of $Z = XY$ under independent Gaussians follow directly from the law of total variance: $\mathbb{E}[XY] = \mu_x\mu_y$ and $\mathrm{Var}[XY] = \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \mu_y^2\sigma_x^2$.

Proposition 3.1 (Product Factor Forward Message)**.** *For the factor $\delta(z - xy)$ with independent inputs $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, the DMA message to Z is $\mathcal{N}(z; m_z, s_z^2)$ with*

$$m_z = \mu_x\mu_y, \qquad s_z^2 = \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \mu_y^2\sigma_x^2. \tag{6}$$

Because the moments (6) are the *exact* moments of $Z = XY$ (not approximations), the only approximation is in the projection of the marginal of Z onto a Gaussian. In the Dirac limit $\sigma_x, \sigma_y \to 0$ both moments converge to those of a point mass at $\mu_x\mu_y$, so $\delta = 0$ and Corollary 2.4 applies.

Backward message via log-normal intermediates. The backward message to X requires the moments of $X = Z/Y$. Direct integration is problematic: the ratio of two Gaussians has no finite mean (the integrand carries a $1/|y|$ singularity). We resolve this using a *log-normal intermediate* technique: for a Gaussian $W \sim \mathcal{N}(\mu_w, \sigma_w^2)$ with $|\mu_w|/\sigma_w \gg 1$ (a sufficient condition for the approximation quality guaranteed by Theorem 3.3), the log-absolute-value $\log|W|$ is approximately normal with mean $\log|\mu_w|$ and variance σ_w^2/μ_w^2 (a first-order delta-method approximation). Under this approximation, $\log|Z/Y| = \log|Z| - \log|Y|$ is the difference of two independent normals, giving a normal distribution whose parameters can be propagated in closed form. Converting back to natural parameters $(\tau_w, \rho_w) = (\mu_w/\sigma_w^2, 1/\sigma_w^2)$ yields explicit moment formulas for X under the joint (Y, Z).

Proposition 3.2 (Product Factor Backward Message)**.** *For the factor $\delta(z - xy)$, the DMA message to X in natural parameters is $\mathcal{N}(\hat\tau_x/\hat\rho_x,\, 1/\hat\rho_x)$ with*

$$\hat\tau_x = \frac{\tau_y\, r\, \tau_z}{D}, \quad \hat\rho_x = \frac{r^2}{D}, \tag{7}$$

where $r := \tau_y^4\rho_z/(\rho_y(\tau_y^2 + \rho_y))$, and $D := \tau_z^2\rho_y + \tau_y^2\rho_z + \rho_y\rho_z$. The message to Y is obtained by symmetry (swap $x \leftrightarrow y$ labels). The derivation is given in Appendix B.4.

Consistency verification. Taking $\rho_y \to \infty$ (Dirac limit on Y) gives $\hat{\tau}_x/\hat{\rho}_x \to \mu_z/\bar{y}$ and $1/\hat{\rho}_x \to \sigma_z^2/\bar{y}^2$, recovering the distribution of $X = Z/\bar{y}$; the derivation is in Appendix B.4.

3.2 Approximation Guarantee for the Product Backward Message

The exact backward message is improper, placing it outside the scope of Theorem 2.3, which requires a proper true message. Theorem 3.3 below is the dedicated complement: it covers this case by establishing a C/r^2 upper bound in the concentrated-input regime (r_y/r_z bounded) where the improper tail is exponentially negligible. Concretely, let $r := \min(r_y, r_z) = \min(|\mu_y|/\sigma_y,\ |\mu_z|/\sigma_z)$; for $r \geq 2$ one constructs a proper *truncated reference* $\tilde{m}^r_{f\to X}$ (Appendix B.5) by restricting to $|Y| \geq |\mu_y|/2$; the excluded input region has probability $\leq \Phi(-r_y/2) = O(e^{-cr^2})$.

Theorem 3.3 (Concentrated-Input Bound for Product Backward Message)**.** *Let $r = \min(|\mu_y|/\sigma_y,\ |\mu_z|/\sigma_z) \geq 2$, assume the ratio r_y/r_z is bounded above and below by a constant $\kappa \geq 1$, and let $\hat{m}_{f\to X}$ be the DMA message of Proposition 3.2. Then*

$$\mathrm{KL}\big[\tilde{m}^r_{f\to X} \,\|\, \hat{m}_{f\to X}\big] \ \leq\ C_\kappa\, r^{-2}, \tag{8}$$

for some constant $C_\kappa > 0$ depending only on κ. The proof is in Appendix B.5.

3.3 The ReLU Factor

Factor definition. The ReLU factor encodes the leaky-ReLU nonlinearity $y = \mathrm{ReLU}_\alpha(x) := \max(0, x) + \alpha \min(0, x)$ for $\alpha \geq 0$:

$$f(x, y) = \delta(y - \mathrm{ReLU}_\alpha(x))\,. \tag{9}$$

Setting $\alpha = 0$ gives the standard ReLU; $\alpha = 1$ gives the identity; $\alpha > 0$ gives a leaky variant. We focus on the leaky case ($\alpha > 0$), which is used throughout the BNN experiments; the standard ReLU ($\alpha = 0$) is discussed only to identify where its backward message becomes improper (Remark B.6).

Forward message. The exact forward message to Y is the pushforward of $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ through $\mathrm{ReLU}_\alpha(\cdot)$. This is a mixture of a truncated Gaussian on $\mathbb{R}_{>0}$ (for $x > 0$, $y = x$) and a scaled truncated Gaussian on $\mathbb{R}_{\leq 0}$ (for $x \leq 0$, $y = \alpha x$); neither piece is Gaussian. We apply moment matching via the doubly-truncated Gaussian moments (whose computation uses the Mills ratio $M(t) := \phi(t)/\Phi(t)$ for the CDF and PDF of the standard normal).

Proposition 3.4 (ReLU Forward Message)**.** *For the factor (9) with $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$, let $u := \frac{\mu_x}{\sigma_x}$, $P := \Phi(u)$, $\phi := \varphi(u)$, $A := \alpha + (1-\alpha)P$, $B := \alpha^2 + (1-\alpha^2)P$. The DMA forward message is $\mathcal{N}\big(y; m_Y, s_Y^2\big)$ with*

$$m_Y = \mu_x A + (1-\alpha)\,\sigma_x\,\phi, \qquad s_Y^2 = (\mu_x^2 + \sigma_x^2)B + (1-\alpha^2)\,\mu_x\,\sigma_x\,\phi - m_Y^2. \tag{10}$$

Backward message. The backward message to X integrates the factor against the incoming message $m_{Y\to f}(y) = \mathcal{N}(\mu_y, \sigma_y^2)$. For $x > 0$ the Jacobian of $y = x$ is 1; for $x \leq 0$ the Jacobian of $y = \alpha x$ is α, contributing a factor $1/\alpha$ to the unnormalised message on that piece. The total mass of the (unnormalised) backward mixture is therefore $\tilde{C}/\alpha$ with $\tilde{C} := \alpha P + Q$, where $v := \mu_y/\sigma_y$, $P := \Phi(v)$, $Q := 1 - P$. Normalising by $\tilde{C}/\alpha$ before matching moments yields a proper Gaussian approximation for all $\alpha > 0$.

Proposition 3.5 (ReLU Backward Message)**.** *For the factor $\delta(y - \mathrm{ReLU}_\alpha(x))$ with $\alpha > 0$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, let $v := \mu_y/\sigma_y$, $P := \Phi(v)$, $Q := 1 - P$, $\phi := \varphi(v)$, and $\tilde{C} := \alpha P + Q$. The DMA backward message is $\mathcal{N}\big(x; m_X, s_X^2\big)$ with*

$$m_X = \frac{\mu_y(\alpha^2 P + Q) + (\alpha^2 - 1)\,\sigma_y\,\phi}{\alpha\,\tilde{C}}, \tag{11}$$

$$s_X^2 = \frac{(\mu_y^2 + \sigma_y^2)(\alpha^3 P + Q) + (\alpha^3 - 1)\,\mu_y\,\sigma_y\,\phi}{\alpha^2\,\tilde{C}} - m_X^2. \tag{12}$$

Consistency verification. In both directions the Dirac limit ($u \to \pm\infty$ forward, $v \to \pm\infty$ backward) recovers $\mathrm{ReLU}_\alpha(\bar{x})$ and $\mathrm{ReLU}_\alpha(\bar{y})^{-1}$ respectively; the derivation is in Appendix B.6. For $\alpha = 0$ the backward message is improper (Appendix B.6).

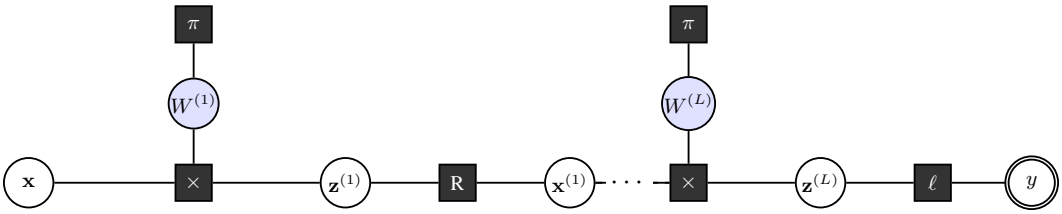


Figure 1: Factor graph of a BNN (schematic; two learnable layers shown). Circles are variable nodes; filled squares are factor nodes. Blue-tinted circles represent the $d_l \times d_{l-1}$ independent scalar weight variables $W_{ij}^{(l)}$ (shown collectively per layer for clarity), each with a Gaussian prior factor π. Each $\times$ node is a *matrix-vector product factor*: shorthand for $d_l \times d_{l-1}$ 1D product factors $\delta\Big(z_{ij} - W_{ij}^{(l)}\, x_j^{(l-1)}\Big)$ (Sec. 3.1) combined with d_l sum factors $z_i^{(l)} = \sum_j z_{ij}^{(l)}$, together implementing the inner product (see Sec. 4.1). R factors are element-wise leaky-ReLU factors (Sec. 3.3); ℓ is the Gaussian likelihood factor (conjugate; $\delta = 0$); the double circle is the observed output y.

4 Application: Bayesian Neural Network Inference

The two DMA factors from Section 3 are sufficient to implement a complete inference algorithm for feedforward BNNs composed of Gaussian, product, sum, and leaky-ReLU factors. We assemble them here, apply the three structural corollaries to the proper-message components, and validate the resulting algorithm (one forward/backward factor sweep per training example, no EP-style inner-loop fixed-point iteration) on a 1D regression task.

4.1 Factor Graph of a Feedforward Network

A feedforward BNN with L layers, input $\mathbf{x} \in \mathbb{R}^{d_0}$, and hidden widths $d_1, \ldots, d_L$ is modelled as

$$p(\mathbf{W}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)}, y) = p(y \mid \mathbf{x}^{(L)}) \prod_{l=1}^{L} p(\mathbf{W}^{(l)})\, \delta\Big(\mathbf{z}^{(l)} - \mathbf{W}^{(l)}\mathbf{x}^{(l-1)}\Big)\; \delta\Big(\mathbf{x}^{(l)} - \mathrm{ReLU}_\alpha\Big(\mathbf{z}^{(l)}\Big)\Big)\,, \tag{13}$$

with $\mathbf{x}^{(0)} = \mathbf{x}$. Each Dirac-delta factor decomposes into independent 1D factors: the matrix-vector product $\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{x}^{(l-1)}$ decomposes into $d_l \times d_{l-1}$ 1D product factors $\delta\Big(z_{ij} - W_{ij}^{(l)}\, x_j^{(l-1)}\Big)$ connected by d_l sum factors $z_i^{(l)} = \sum_j z_{ij}^{(l)}$ that implement the inner product; the activation $\mathbf{x}^{(l)} = \mathrm{ReLU}_\alpha\big(\mathbf{z}^{(l)}\big)$ splits into d_l independent ReLU factors $\delta\Big(x_i^{(l)} - \mathrm{ReLU}_\alpha\Big(z_i^{(l)}\Big)\Big)$.

Figure 1 illustrates the factor graph schematically; each $\times$ node is a matrix-vector product factor composed of $d_l \times d_{l-1}$ 1D product factors and d_l sum factors. The three factor types present (Gaussian prior factors, product factors, and ReLU factors) carry conjugate, DMA (Prop. 3.1), and DMA (Prop. 3.4) messages respectively. By Corollary 2.4, all prior and linear-copy factors satisfy the consistency condition; ReLU factors are the dominant source of nonzero δ for uncertain inputs.

4.2 Inference Algorithm: One Forward/Backward Sweep per Example

Corollary 2.5 guarantees that no EP-style inner-loop fixed-point iteration between individual message computations is required: each factor-to-variable message is computed exactly once per sweep. We fix the order as a single forward sweep followed by a single backward sweep per training example, mirroring the computation graph of a standard neural network. The prior and likelihood factors produce exact Gaussian messages (Winn & Bishop, 2005); the non-conjugate factors use Propositions 3.1–3.5. Algorithm 1 gives the complete training procedure (split across two pages); each mini-batch maintains one stored factor-to-weight message that is replaced (not accumulated) on each pass, so weight beliefs are updated by EP-style message replacement rather than sequential accumulation. Algorithm 1 runs in $O(\sum_l d_l d_{l-1})$ time per example, matching the asymptotic complexity of a standard neural network forward-backward pass with a modest constant-factor overhead ($1.8\times$ per epoch), and requires only two scalar parameters (τ, ρ) per weight belief plus one stored factor message per mini-batch per weight.

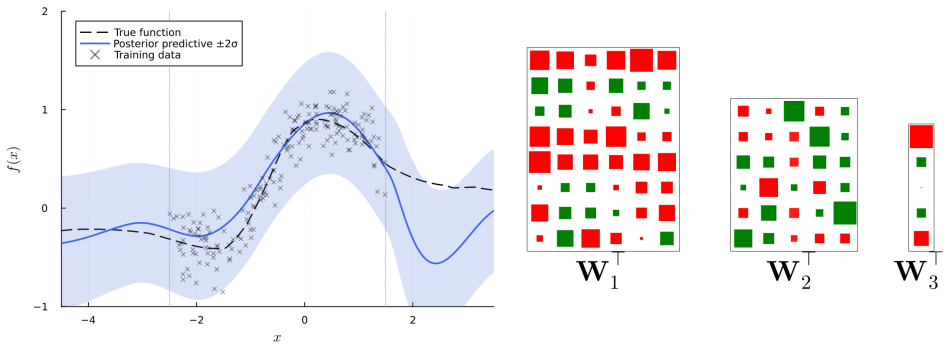


Figure 2: *Left*: Posterior predictive mean (solid) and $\pm 2\sigma$ intervals (shaded) versus the true data-generating function (dashed), where $\sigma = \sqrt{\mathrm{Var}[f(x)] + \beta^2}$ combines posterior variance with observation noise (β=0.2). Training points shown as crosses ($N = 200$, $x \in [-2.5, 1.5]$); dashed verticals mark the training boundaries. *Right*: Hinton diagram of posterior weight beliefs after training. Square area $\propto$ posterior mean magnitude; transparency $\propto$ posterior variance (opaque = certain).

4.3 Illustrative Experiment: 1D Regression

Setup. We evaluate Algorithm 1 on a scalar regression task under a correctly specified model: the ground truth f is itself a draw from the BNN prior, which ensures that the model's parameters can be learned at all, a standard simulation-study design that isolates inference quality from model misspecification. We generate $N = 200$ noise-corrupted observations at inputs drawn uniformly from $[-2.5, 1.5]$ and run DMA to recover the posterior. A fixed 7-component feature map φ is prepended and standardised; a two-hidden-layer leaky-ReLU network ($d = 6, 5$) with Gaussian likelihood ($\beta = 0.2$) is trained for up to 200 epochs with mini-batches of 20. Full architecture and hyperparameter details (prior scale σ_0, likelihood β, activation slopes α) are in Appendix E.

Results. Figure 2 shows the posterior predictive distribution and the posterior weight beliefs after training. The predictive mean tracks the true function closely within the training region $[-2.5, 1.5]$; the $\pm 2\sigma$ intervals widen in the extrapolation regions, where the training data provides no information, a qualitatively appropriate epistemic uncertainty that grows where the model is uninformed.

Discussion. The experiment illustrates the behavior predicted by all three corollaries: no EP-style inner-loop iteration is needed (Corollary 2.5), all weight variances remain positive throughout (Corollary 2.6), and the posteriors are non-trivial distributions because linear factors satisfy the consistency condition and contribute small δ (Corollary 2.4). A comparison with Adam (Kingma & Ba, 2015), AdamW (Loshchilov & Hutter, 2019) at four weight-decay values, and the diagonal Laplace approximation (MacKay, 1992) (Appendix E) shows, on this correctly specified 1D task, DMA requires few epochs (median epoch 16) with no gradient learning-rate hyperparameter; in the example, over 20 seeds DMA achieves a median extrapolation NLL of 0.80 versus 4.05 for AdamW (Appendix E.3). Both methods show similar calibration error over 20 seeds (DMA median $\Delta = -0.09$, diagonal Laplace $\Delta = -0.10$); neither is systematically overconfident (Appendix E.7). Under model mismatch (data generated by a wider network), DMA's posterior predictive continues to widen outside the training range while Adam provides no epistemic uncertainty; see Appendix F.

Larger Networks. While the first experiment uses a small network for demonstration, Appendix G evaluates DMA on a larger 1 932-weight, four-output network ($6 \to 6 \to 12 \to 48 \to 24 \to 4$, $N = 1500$). DMA obtains competitive results in 3 epochs (0.72 s total), compared with 1.7 s for Adam ($\eta = 0.01$, requiring $\sim$100 epochs). We used 100 mini-batches with the default settings in Appendix E.1.

5 Related Work

Marginal-error bounds from message perturbations. Ihler et al. (2005) bound marginal changes under total-variation perturbations via a graph-global contraction constant. Theorem 2.3 is complementary: it handles projected messages where $m \notin \mathcal{Q}$, is edge-local, and requires no contraction condition.

Expectation propagation and variants. EP (Minka, 2001) recovers outgoing messages by projecting the marginal and dividing out the cavity (eq. 20). Power EP (Minka, 2004), α-EP (Minka, 2005), and Stochastic EP (Li et al., 2015) generalise the projection or reduce memory, but all retain cavity division and its pathologies. Hernández-Lobato & Adams (2015) apply EP to a BNN, requiring per-example cavity computation and iterative sweeps; DMA eliminates both (Definition 2.1).

Assumed density filtering. ADF (Lauritzen, 1992; Opper & Winther, 1999) also avoids cavity division, but for a structural reason: it is a forward-only sequential algorithm that absorbs each observation into a running prior and never performs a backward sweep. Weight beliefs are therefore never updated via backward messages — the step DMA's product backward message (Section 3.1) is designed to handle. Table 1 (Appendix A) compares ADF, EP, VMP, and DMA.

Variational inference and approximate BNNs. VMP (Winn & Bishop, 2005) and ELBO-based methods use the same marginal-first division as EP; for deterministic factors this collapses the posterior to a Dirac delta (Appendix A). Bayes by Backprop (Blundell et al., 2015), MC Dropout (Gal & Ghahramani, 2016), and SWAG (Maddox et al., 2019) approximate the posterior via ELBO or SGD trajectory statistics; none maintains explicit factor-graph message structure.

Laplace approximation and gradient-based uncertainty. Diagonal Laplace (MacKay, 1992) fits a Gaussian at the MAP post-hoc. IVON (Shen et al., 2024) tracks a diagonal natural-gradient estimate online; it propagates uncertainty through gradient statistics rather than factor-graph messages and provides no backward product message to weight beliefs.

Message-passing systems. TrueSkill (Herbrich et al., 2006) demonstrates EP at scale on conjugate factors only, so the Dirac-delta collapse does not arise. Infer.NET (Minka et al., 2018) provides a general EP/VMP engine with pluggable message operators, illustrating the demand for modular approximate-message infrastructure.

6 Discussion and Conclusions

Summary. We have introduced direct message approximation (DMA), a framework for approximate inference on factor graphs that replaces the marginal-projection step of EP with a direct approximation of the outgoing message. The guarantees form a two-tier structure. **Theorem 2.3** (*proper* messages, any graph): bounds the marginal KL at any edge by $O(\delta)$ with an edge-local, topology-independent prefactor; three corollaries establish: no EP-style inner-loop iteration, no negative-precision messages, and Dirac-input consistency at all linear factors. **Theorem 3.3** (*improper* product-backward message, concentrated-input regime): establishes a C/r^2 upper bound where Theorem 2.3 does not apply; Appendix C.3 confirms this rate empirically against importance-sampling reference marginals (Figure 5).

The BNN instantiation shows that DMA's structural guarantees carry through to larger models without sacrificing epistemic quality for scalability. DMA fits the training data well and requires no gradient learning-rate hyperparameter, avoiding the manual learning-rate tuning that gradient-based methods require. Extrapolation uncertainty is as well calibrated as diagonal Laplace (limited to smaller models); the widening intervals reflect genuine epistemic uncertainty propagated structurally through the factor graph, not a seed artefact. Under model mismatch, where no weight setting can fit the true function exactly, DMA's posterior captures residual uncertainty and produces useful predictive intervals beyond the training range; Adam, lacking uncertainty quantification, extrapolates without any such signal. The procedure scales to substantially larger networks without algorithmic change and at wall-clock time comparable to Adam (Appendices E–G).

Limitations. The master theorem requires proper outgoing messages; factors whose exact messages are improper (such as the product-backward) need separate treatment beyond the general recipe. DMA maintains a fully factorised Gaussian belief; block-diagonal extensions are compatible in principle but require moment formulas under non-diagonal inputs.

Future directions. Bounding accumulated error after $K > 1$ sweeps on cyclic graphs is the most pressing open problem; closing this gap would give end-to-end convergence guarantees. The construction recipe extends to any factor with closed-form moment formulas, making the factor library a matter of mathematical derivation rather than framework change. Student-t and mixture-of-Gaussians messages are natural next targets for heavier-tailed or multimodal posteriors.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and minor implementation support, including assistance with code-related tasks during development. We have not used generative AI tools to generate the scientific contributions, theoretical results, proofs, experimental findings, or interpretations presented in this work. We have reviewed and verified all AI-assisted text and implementation changes, including checking the correctness of any AI-assisted code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not identify any ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We provide an anonymous Git repository containing the implementation of DMA, the experimental configurations, and scripts used to reproduce the reported results. The main paper and Appendix B provide the assumptions, derivations, and complete proofs of the theoretical results, while Appendix D describes the experimental setups and hyperparameters. Appendix E provides the experimental procedures and reference computations used to validate the theoretical error bounds. Together, these materials provide the information needed to reproduce both the theoretical and empirical results reported in this work.

All results were produced on a consumer laptop (Intel Core Ultra 5 225U, 32 GB RAM, single-threaded using Julia).

An anonymous code repository can be found here:

https://anonymous.4open.science/r/DMA-Anom-0484/README.md

REFERENCES

Shun-ichi Amari. *Differential-Geometric Methods in Statistics*, volume 28 of *Lecture Notes in Statistics*. Springer, 1985.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2006.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Alexander T. Ihler, John W. Fisher, and Alan S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, 2005.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust Gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12:3227–3257, 2011. URL https://www.jmlr.org/papers/volume12/jylanki11a/jylanki11a.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

Steffen L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

David J. C. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1992.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

Thomas P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Thomas P. Minka. Power EP. In *Microsoft Research Technical Report MSR-TR-2004-149*, 2004.

Tom Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, 2005.

Tom Minka, John Winn, John Guiver, David Knowles, and Yordan Zaykov. Infer.NET 0.3. Technical report, Microsoft Research, 2018. Cambridge, UK. http://dotnet.github.io/infer.

Manfred Opper and Ole Winther. A mean field algorithm for Bayes learning in large feed-forward neural networks. In *Advances in Neural Information Processing Systems 11 (NIPS)*, 1999.

Yuesong Shen, Nico Daheim, Bai Cong, Peter Nickl, Gian Maria Marconi, Clement Bazan, Rio Yokota, Iryna Gurevych, Daniel Cremers, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Variational learning is effective for large deep networks. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2024.

John M. Winn and Christopher M. Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(23):661–694, 2005.

Appendix Overview

A **Background** — factor graphs, sum-product algorithm, α-divergence family, EP, ADF, and VMP as marginal-based methods, and a comparison table.

B **Proofs** — all theorem and proposition proofs, including the master theorem (Thm. 2.3), product-factor constructions (Props. 3.1–3.2 and Thm. 3.3), ReLU-factor constructions, and a theorem-coverage table by factor and direction.

C **Empirical validation** — numerical verification of the $O(\delta + \sqrt{\delta})$ master-theorem rate and the $O(1/r^2)$ concentrated-input rate against importance-sampling reference marginals.

D **Inference algorithm** — complete pseudocode for Algorithm 1.

E **Experimental details** — feature map, architecture, hyperparameters, extended Adam, AdamW, and diagonal-Laplace comparisons, IVON sweep, EP exclusion rationale, and calibration curves.

F **Model mismatch** — 20-seed experiment comparing DMA and Adam when the inference model is narrower than the data-generating network.

G **Larger-scale BNN** — DMA on a 1 932-weight, four-output network; baseline-choice rationale; calibration at scale.

A Background (Notation and Review)

A.1 Factor Graphs and the Sum-Product Algorithm

A *factor graph* is a bipartite graph that encodes the factorisation of a joint density over n variables $X_1, \dots, X_n$ into m local potentials:

$$p(x_1, \dots, x_n) = \prod_{i=1}^{m} f_i\big(\mathbf{x}_{\mathrm{ne}(f_i)}\big), \tag{14}$$

where $\mathrm{ne}(f_i) \subseteq \{1, \dots, n\}$ is the neighbourhood of factor f_i and $\mathrm{ne}(X_j) \subseteq \{1, \dots, m\}$ is the neighbourhood of variable X_j.

On a factor *tree*, the sum-product algorithm computes the exact marginal p_{X_j} of every variable via four message equations (Kschischang et al., 2001):

$$p_{X_j}(x_j) = \prod_{i \in \mathrm{ne}(X_j)} m_{f_i \to X_j}(x_j), \tag{15}$$

$$m_{f_i \to X_j}(x_j) = \int f_i\big(\mathbf{x}_{\mathrm{ne}(f_i)}\big) \prod_{k \in \mathrm{ne}(f_i) \setminus \{j\}} m_{X_k \to f_i}(x_k)\, \mathrm{d}\mathbf{x}_{\mathrm{ne}(f_i)\setminus\{j\}}, \tag{16}$$

$$m_{X_j \to f_i}(x_j) = \prod_{k \in \mathrm{ne}(X_j) \setminus \{i\}} m_{f_k \to X_j}(x_j). \tag{17}$$

The factor-to-variable message (16) integrates the factor against all incoming variable messages *except* the one at the target edge; similarly, the variable-to-factor message (17) is the product of all incoming factor messages except at the target edge. Multiplying the two messages on any edge recovers the *marginal-edge identity*:

$$p_{X_j}(x_j) = m_{f_i \to X_j}(x_j) \cdot m_{X_j \to f_i}(x_j). \tag{18}$$

Conjugacy constraint. The integral in (16) is in closed form only when f_i is conjugate to the incoming message family $\mathcal{Q}$. For Gaussian messages, linear factors $\delta\big(x_j - \mathbf{a}^\top \mathbf{x}_{-j}\big)$ and Gaussian-likelihood factors are conjugate; the scalar product factor $\delta(z - xy)$ and activation factors such as ReLU and softmax are not, and (16) does not yield a Gaussian in closed form.

A.2 The Projection Step and the α-Divergence Family

When the exact message (16) is intractable, it is replaced by a projection onto a tractable family $\mathcal{Q}$. The α-divergence (Amari, 1985; Minka, 2005)

$$D_\alpha(p \,\|\, q) = \frac{1}{\alpha(1-\alpha)}\Big(1 - \int p(x)^\alpha\, q(x)^{1-\alpha}\, \mathrm{d}x\Big), \quad \alpha \in \mathbb{R},$$

unifies many classical algorithms: $\lim_{\alpha \to 1} D_\alpha = \mathrm{KL}[p \,\|\, q]$ (forward KL, used by EP) and $\lim_{\alpha \to 0} D_\alpha = \mathrm{KL}[q \,\|\, p]$ (reverse KL, used by VMP). The projection step in both algorithms relies on the following standard result.

Theorem A.1 (Moment-Matching Minimiser of KL)**.** *Let p be an arbitrary density and $\mathcal{Q} = \{q(\cdot \mid \boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^\top \mathbf{T}(\cdot) - A(\boldsymbol{\theta}))\}$ an exponential family with sufficient statistic $\mathbf{T}$. The minimiser $q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}[p \,\|\, q]$ satisfies*

$$\mathbb{E}_{q^*}[\mathbf{T}(X)] = \mathbb{E}_p[\mathbf{T}(X)]. \tag{19}$$

For Gaussian $\mathcal{Q}$ the condition (19) reduces to matching the mean and variance of p. Theorem A.1 will serve as the projection step in both EP and in the DMA construction of Section 2.2.

A.3 Prior Approximate Inference Methods

Expectation propagation. EP (Minka, 2001) approximates each factor-to-variable message in three steps.

1. Form the *true unnormalised marginal* $p_{X_j} = m_{f_i \to X_j} \cdot m_{X_j \to f_i}$ using the exact integral (16) (combined with (18)).
2. Project p_{X_j} onto $\mathcal{Q}$ via Theorem A.1, obtaining $\hat{p}_{X_j} \in \mathcal{Q}$.
3. Recover the approximate message by dividing out the incoming message:

$$\hat{m}_{f_i \to X_j}(\cdot) = \hat{p}_{X_j}(\cdot) \;/\; m_{X_j \to f_i}(\cdot). \tag{20}$$

The approximation is exact when $p_{X_j} \in \mathcal{Q}$; otherwise EP introduces a local error and must iterate to a fixed point.

Variational message passing. VMP (Winn & Bishop, 2005) instead maximises the evidence lower bound $\mathcal{L}(\hat{p}) = \mathbb{E}_{\hat{p}}[\log p - \log \hat{p}]$ under a fully factorised approximation $\hat{p}(\mathbf{x}) = \prod_k \hat{p}_{X_k}(x_k)$, which is equivalent to minimising $\mathrm{KL}[\hat{p} \,\|\, p]$. Coordinate ascent on $\hat{p}_{X_k}$ yields

$$\hat{p}_{X_k}(x_k) \propto \exp\!\big(\mathbb{E}_{\hat{p}_{-k}}[\log f_i(\mathbf{x}_{\mathrm{ne}(f_i)})]\big), \tag{21}$$

where the expectation is over all neighbours of f_i except X_k. When $\log f_i$ decomposes linearly in each argument separately, this produces a message $\hat{m}_{f_i \to X_k} \propto \hat{p}_{X_k}$; again the message is recovered by dividing out the incoming message at the same edge, so VMP also obeys the form (20).

Assumed-density filtering. ADF (Lauritzen, 1992; Opper & Winther, 1999) processes observations sequentially, projecting the posterior onto $\mathcal{Q}$ after each observation and using it as the prior for the next; no stored factor message needs to be divided out. ADF therefore also avoids cavity division, but for a structural reason: it is an online, forward-only algorithm that approximates the *marginal* at each step (Table 1). Crucially, standard ADF's forward-only architecture provides no mechanism for propagating output likelihood information backward to update upstream weight beliefs — the technically difficult step that DMA's backward product message is designed to handle.

Pathologies of the marginal-first design. EP and VMP approximate $\hat{p}_{X_j}$ and recover the message by division (1), inducing three pathologies. *(i) Iteration (EP and VMP).* The division makes $\hat{m}_{f_i \to X_j}$ depend on $m_{X_j \to f_i}$, so updating one edge invalidates neighbouring messages; convergence requires repeated sweeps. *(ii) Negative precision (Gaussian EP).* When $\hat{p}_{X_j}$ is wider than $m_{X_j \to f_i}$, the ratio (20) yields a Gaussian with negative precision, an invalid distribution that propagates downstream. *(iii) Dirac-delta collapse (VMP).* For a deterministic factor $f_i(\mathbf{x}) = \delta(x_j - g(\mathbf{x}_{-j}))$, the ELBO update (21) forces $\hat{p}_{X_j}$ to a Dirac delta; since every parameterised layer in the factorisation used here is such a factor, VMP cannot propagate non-trivial posterior variance backward to the weights. Section 2 introduces DMA, which avoids this design by approximating the message directly.

Table 1: Comparison of approximate inference frameworks on properties relevant to BNN inference. Below the first row, ✓ indicates a desirable property. *Approximates*: whether the method targets the factor-to-variable message directly or approximates the marginal at each edge. *No cavity division*: whether updates avoid dividing out a stored incoming message (eq. 20). *No inner-loop iteration*: whether a single forward/backward sweep per example suffices, with no per-example fixed-point iteration. *Backward weight updates*: whether weight beliefs are updated via a backward message sweep. *Consistency axiom / edge-local KL bound*: see Definition 2.1 and Theorem 2.3.

	ADF	EP	VMP	**DMA (ours)**
Approximates	marginal	marginal	marginal	**message**
Backward weight updates	×	✓	✓	✓
No cavity division	✓	×	×	✓
No inner-loop iteration	✓	×	×	✓
Consistency axiom	×	×	×	✓
Edge-local KL bound	×	×	×	✓

B PROOFS

B.1 PROOF OF THEOREM A.1 (MOMENT-MATCHING MINIMISER OF KL)

Proof. For $q(\cdot \mid \boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^\top \mathbf{T}(\cdot) - A(\boldsymbol{\theta}))$ in the exponential family:

$$\mathrm{KL}[p \,\|\, q] = \int p(x) \log p(x)\, \mathrm{d}x - \int p(x) \big(\boldsymbol{\theta}^\top \mathbf{T}(x) - A(\boldsymbol{\theta})\big)\, \mathrm{d}x = -H(p) + A(\boldsymbol{\theta}) - \boldsymbol{\theta}^\top \mathbb{E}_p[\mathbf{T}].$$

Differentiating with respect to $\boldsymbol{\theta}$ and setting to zero:

$$\nabla_{\boldsymbol{\theta}} \mathrm{KL}[p \,\|\, q] = \nabla A(\boldsymbol{\theta}) - \mathbb{E}_p[\mathbf{T}] = \mathbf{0}.$$

For exponential families, $\nabla A(\boldsymbol{\theta}) = \mathbb{E}_{q(\cdot|\boldsymbol{\theta})}[\mathbf{T}]$ (the moment-generating identity), so the stationary condition is exactly $\mathbb{E}_{q^*}[\mathbf{T}(X)] = \mathbb{E}_p[\mathbf{T}(X)]$. The KL is convex in $\boldsymbol{\theta}$ (since A is log-partition and hence convex), so the stationary point is the unique minimum. For Gaussian $\mathcal{Q}$, $\mathbf{T}(x) = (x, x^2)$ and the condition reduces to matching the first two moments. □

B.2 PROOF OF THEOREM 2.3 (DMA MASTER THEOREM)

Proof. Write $m := m_{f \to X_j}$, $\hat{m} := \hat{m}_{f \to X_j}$, Z, $\hat{Z}$, δ as in the theorem statement; assume $\|m_{X_j \to f}\|_\infty < \infty$. For $a, b > 0$ define the scalar Bregman divergence $D(a, b) := a \log(a/b) - a + b \geq 0$, with the same generator $\phi(t) = t \log t$ as the KL. The key identity is the *exact decomposition*

$$Z\, \mathrm{KL}\Big[\tfrac{p_{X_j}}{Z} \,\|\, \tfrac{\hat{p}_{X_j}}{\hat{Z}}\Big] \;=\; \int m_{X_j \to f}(x)\, D\big(m(x), \hat{m}(x)\big)\, \mathrm{d}x \;-\; D(Z, \hat{Z}). \tag{22}$$

Proof of (22). Expand the left side: $Z\, \mathrm{KL}\Big[p/Z \,\|\, \hat{p}/\hat{Z}\Big] = \int m_{X_j \to f}\, m \log(m/\hat{m}) + Z \log(\hat{Z}/Z)$. Adding and subtracting $\int m_{X_j \to f}(m - \hat{m}) = Z - \hat{Z}$ gives

$$\int m_{X_j \to f}(x) \underbrace{\big[m(x) \log(m(x)/\hat{m}(x)) - m(x) + \hat{m}(x)\big]}_{D(m(x), \hat{m}(x))}\, \mathrm{d}x + \underbrace{(Z - \hat{Z}) + Z \log(\hat{Z}/Z)}_{-D(Z, \hat{Z})},$$

where the second group equals $-D(Z, \hat{Z}) = -(Z \log(Z/\hat{Z}) - Z + \hat{Z})$. □

Since $D(Z, \hat{Z}) \geq 0$, discarding it and dividing by Z gives

$$\mathrm{KL}\Big[\tfrac{p_{X_j}}{Z} \,\|\, \tfrac{\hat{p}_{X_j}}{\hat{Z}}\Big] \;\leq\; \frac{1}{Z} \int m_{X_j \to f}(x)\, D\big(m(x), \hat{m}(x)\big)\, \mathrm{d}x.$$

Since $D(m(x), \hat{m}(x)) \geq 0$ and $m_{X_j \to f} \geq 0$, bounding $m_{X_j \to f}(x) \leq \|m_{X_j \to f}\|_\infty$ is valid. Using $\int D(m, \hat{m})\, \mathrm{d}x = \int m \log(m/\hat{m})\, \mathrm{d}x = \delta$ (the $-m + \hat{m}$ terms integrate to zero since $\int m\, \mathrm{d}x = \int \hat{m}\, \mathrm{d}x = 1$) gives the stated bound. □

B.3 PROOFS OF COROLLARIES 2.4, 2.5, AND 2.6

Proof of Corollary 2.4. Take each incoming message as $\mathcal{N}(\bar{x}_k, \sigma_k^2)$ with $\sigma_k > 0$. For every $\sigma_k > 0$ both $m_{f \to X_j}$ and $\hat{m}_{f \to X_j}$ are normalizable densities, so the KL is well-defined and finite.

By the construction recipe (Section 2.2), $\hat{m}_{f \to X_j} = \mathcal{N}(\mu_\sigma, \sigma_\sigma^2)$ where $\mu_\sigma = \mathbb{E}[g(\mathbf{X}_{-j})]$ and $\sigma_\sigma^2 = \mathrm{Var}[g(\mathbf{X}_{-j})]$ are the first two moments of the exact outgoing message. Since the DMA matches these moments exactly, for any distribution m and its moment-matched Gaussian projection $\hat{m}$:

$$E_m[\log \hat{m}(X)] = -\tfrac{1}{2} \log(2\pi\sigma_\sigma^2) - \tfrac{1}{2} = -H(\hat{m}),$$

which gives the identity

$$\mathrm{KL}\big[m_{f \to X_j} \,\|\, \hat{m}_{f \to X_j}\big] = H(\hat{m}_{f \to X_j}) - H(m_{f \to X_j}).$$

The C^2 assumption in a neighbourhood of $\bar{\mathbf{x}}_{-j}$ provides the second-order Taylor expansion

$$g(\bar{\mathbf{x}}_{-j} + \mathbf{h}) = g(\bar{\mathbf{x}}_{-j}) + \nabla g(\bar{\mathbf{x}}_{-j})^\top \mathbf{h} + \tfrac{1}{2} \mathbf{h}^\top \nabla^2 g(\bar{\mathbf{x}}_{-j})\, \mathbf{h} + o(\|\mathbf{h}\|^2).$$

The nonzero linear term is $O(\|\sigma\|)$; the nonlinear remainder is $O(\|\sigma\|^2)$, hence asymptotically negligible relative to the linear term. Combined with tail regularity, this makes $H(m_{f\to X_j})$ asymptotically determined by the linearisation, giving $H(m_{f\to X_j}) = \frac{1}{2}\log(2\pi e\,\sigma_\sigma^2) + o(1)$ and hence $\mathrm{KL}\big[m_{f\to X_j} \,\|\, \hat{m}_{f\to X_j}\big] \to 0$. We verify this explicitly factor-class by factor-class.

Conjugate factors (Gaussian prior, Gaussian likelihood): the exact message is already Gaussian, so $m_{f\to X_j} = \hat{m}_{f\to X_j}$ and $\delta = 0$ for all σ_k.

Product forward (Proposition 3.1): $g(x, y) = xy$ is C^∞ everywhere with $\nabla g = (y, x)^\top \neq \mathbf{0}$ when $(\mu_x, \mu_y) \neq (0, 0)$. The DMA computes the exact moments of $Z = XY$; m is a variance-gamma distribution whose entropy satisfies $H(m) = \frac{1}{2}\log\sigma_z^2 + c + o(1)$ as $\sigma_x, \sigma_y \to 0$, matching the leading term of $H(\hat{m})$ with the same constant c, so $\delta = H(\hat{m}) - H(m) \to 0$.

ReLU forward and backward (Propositions 3.4–3.5): leaky-ReLU is piecewise-C^∞ with a single non-smooth point at the origin. When the concentration point $\bar{x}$ is away from the origin, the C^2 argument applies directly. At the origin, the DMA uses exact truncated-Gaussian moments; Mills ratio bounds give $\sigma_\sigma^2 \to 0$ and $H(\hat{m}) - H(m) \to 0$ via the explicit moment formulas.

Hence $\delta \to 0$ in all cases and the bound (4) vanishes. □

Proof of Corollary 2.5. The recipe in Section 2.2 integrates only the incoming messages at neighbours $\{X_k\}_{k\neq j}$; $m_{X_j\to f}$ does not appear. Therefore updating $\hat{m}_{f\to X_j}$ does not change $m_{X_j\to f}$, and no re-computation of neighbouring messages is triggered. □

Proof of Corollary 2.6. The moment-matching step minimises $\mathrm{KL}[p \,\|\, \hat{m}]$ over $\mathcal{Q}$. For the Gaussian family the minimiser has variance $\mathrm{Var}_p[X] \geq 0$ whenever p has finite second moment. If $\mathrm{Var}_p[X] > 0$ the message is a proper Gaussian with strictly positive precision $\rho = 1/\mathrm{Var}_p[X] > 0$. In the Dirac limit ($\sigma_k \to 0$), $\mathrm{Var}_p[X] \to 0$ and the precision $\rho \to \infty$; the message degenerates to a Dirac delta, which is the correct limiting message (Corollary 2.4). At no point is ρ negative. □

B.4 Proofs of Propositions 3.1 and 3.2 (Product Factor DMA)

Proof of Proposition 3.1. For independent $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, the first two moments of $Z = XY$ follow directly from independence and the law of total variance:

$$\begin{aligned}
\mathbb{E}[Z] &= \mathbb{E}[X]\mathbb{E}[Y] = \mu_x\mu_y, \\
\mathbb{E}[Z^2] &= \mathbb{E}[X^2]\mathbb{E}[Y^2] = (\mu_x^2 + \sigma_x^2)(\mu_y^2 + \sigma_y^2), \\
\mathrm{Var}[Z] &= \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 = \sigma_x^2\sigma_y^2 + \mu_x^2\sigma_y^2 + \mu_y^2\sigma_x^2.
\end{aligned}$$

The DMA is the moment-matching Gaussian projection via Theorem A.1. □

Proof of Proposition 3.2 (sketch). We compute the first two moments of $X = Z/Y$ under independent $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$ and $Z \sim \mathcal{N}(\mu_z, \sigma_z^2)$. The ratio distribution has no finite moments under the Gaussian directly (the $1/|y|$ singularity at the origin diverges), so we use a log-normal intermediate.

Lemma B.1 (Log-Normal Approximation)**.** *For $W \sim \mathcal{N}(\mu_w, \sigma_w^2)$ with $|\mu_w|/\sigma_w \gg 1$, the random variable $\log|W|$ is approximately $\mathcal{N}(\log|\mu_w|,\, \sigma_w^2/\mu_w^2)$ by the first-order delta method (Taylor expansion of $\log|\cdot|$ around μ_w).*

Apply Lemma B.1 to Y and Z, treating both as log-normal. Since $\log|Z/Y| = \log|Z| - \log|Y|$, the difference of two independent normals is normal with mean $m = \log|\mu_z| - \log|\mu_y|$ and variance $v = \sigma_z^2/\mu_z^2 + \sigma_y^2/\mu_y^2$.

For a log-normal random variable W with $\log|W| \sim \mathcal{N}(m, v)$, the exact log-normal moments are $\mathbb{E}[W] = \mathrm{sgn}(\mu_z/\mu_y)\cdot e^{m+v/2}$ and $\mathbb{E}[W^2] = e^{2m+2v}$ (using independence of Z and Y). The variance is $\mathrm{Var}[X] = e^{2m+v}(e^v - 1)$.

Converting back to natural parameters $\tau_w = \mu_w/\sigma_w^2$, $\rho_w = 1/\sigma_w^2$, so $m = \log|\tau_z\rho_y/(\tau_y\rho_z)|$ and $v = \rho_z/\tau_z^2 + \rho_y/\tau_y^2$, and algebraic simplification yields the formulas in Proposition 3.2.

Consistency verification. In the limit $\sigma_y \to 0$ ($\rho_y \to \infty$, $\tau_y/\rho_y \to \bar{y}$): $m \to \log|\mu_z/\bar{y}|$ and $v \to \sigma_z^2/\mu_z^2$. Then $\mu_X = e^{m+v/2} \to |\mu_z/\bar{y}| \cdot e^{\sigma_z^2/(2\mu_z^2)} \to \mu_z/\bar{y}$ as $\sigma_z \to 0$; this matches $g(\bar{y}) = \bar{y}$ applied to the constraint $x = z/y$. The direct arithmetic verification of the formula $\hat{\tau}_x/\hat{\rho}_x \to \mu_z/\bar{y}$ and $1/\hat{\rho}_x \to \sigma_z^2/\bar{y}^2$ was given in Section 3.1. □

B.5 Proof of Theorem 3.3 (Concentrated-Input Bound for Product Backward Message)

Remark B.2 (Improper Backward Message)**.** *The exact sum-product backward message to X is*

$$m_{f\to X}(x) = \frac{1}{|x|}\,\mathcal{N}\!\left(\frac{\mu_z}{x};\,\mu_y,\,\sigma_y^2 + \frac{\sigma_z^2}{x^2}\right), \tag{23}$$

which is not normalisable: as $|x| \to \infty$ the Gaussian factor approaches a positive constant and the $1/|x|$ prefactor yields a divergent integral. The DMA (Proposition 3.2) is therefore an approximation to an improper distribution; its moments are the moments of the ratio variable $X = Z/Y$ under the joint on (Y, Z), which is always proper. In the Dirac limit the $1/|x|$ prefactor and the implicit $|x|$ factor from the Gaussian normalisation cancel, recovering the correct point-mass message.

The *truncated reference* excludes the non-normalisable tail of the exact backward message (23):

$$\tilde{m}^r_{f\to X}(x) \;\propto\; \int_{|y|\,\geq\,|\mu_y|/2} \mathcal{N}(xy;\,\mu_z,\sigma_z^2)\,\mathcal{N}(y;\,\mu_y,\sigma_y^2)\,\mathrm{d}y, \tag{24}$$

which is proper for $r \geq 2$ since the $1/|x|$ singularity is integrable when Y is bounded away from zero. For $\mu_y > 0$, the excluded probability is $P(|Y| < \mu_y/2) = \Phi(-r_y/2) - \Phi(-3r_y/2) \leq \Phi(-r_y/2)$, which decays faster than any polynomial in r.

Outline of the proof The proof of Theorem 3.3 applies the Pythagorean identity (Theorem A.1) to decompose $\mathrm{KL}(\tilde{m}^r_{f\to X}\|\hat{m}_{f\to X})$ into two terms, each $\leq C_\kappa/r^2$:

- *Non-Gaussianity* of the truncated ratio $\tilde{m}^r_{f\to X}$ relative to its moment-matched Gaussian N_X^* (Lemma B.5): bounded via the KL chain rule and data-processing inequality, exploiting the fact that $X = Z/Y$ is exactly Gaussian conditionally on $Y = y$.
- *Moment error* of the log-normal DMA approximation relative to N_X^* (Lemmas B.3–B.4): controlled by a second-order Taylor expansion of Z/Y around (μ_z, μ_y), with the Gaussian-to-log-normal KL as an intermediate.

The three lemmas below establish the ingredients in order.

Lemma B.3 (Gaussian–log-normal KL bound)**.** *Let $W \sim \mathcal{N}(\mu_w, \sigma_w^2)$ with $r_w := |\mu_w|/\sigma_w \geq 2$, and let $\mathrm{LN}(\mu_\ell, \sigma_\ell^2)$ be the log-normal whose mean and variance match those of W (i.e. $\sigma_\ell^2 = \log(1 + 1/r_w^2)$, $\mu_\ell = \log|\mu_w| - \sigma_\ell^2/2$). Then*

$$\mathrm{KL}\Big(\mathrm{LN}(\mu_\ell, \sigma_\ell^2)\,\Big\|\,\mathcal{N}(\mu_w, \sigma_w^2)\Big) = \frac{3}{4r_w^2} + O\!\left(\frac{1}{r_w^4}\right). \tag{25}$$

Proof. Let $L \sim \mathrm{LN}(\mu_\ell, \sigma_\ell^2)$ with matched moments $\mathbb{E}[L] = |\mu_w|$ and $\mathrm{Var}[L] = \sigma_w^2$. The cross-entropy of L under $\mathcal{N}(\mu_w, \sigma_w^2)$ equals the entropy of $\mathcal{N}$ by moment matching:

$$H(L, \mathcal{N}(\mu_w, \sigma_w^2)) = -\mathbb{E}_L[\log \mathcal{N}(X; \mu_w, \sigma_w^2)] = \tfrac{1}{2}\log(2\pi\sigma_w^2) + \tfrac{\mathrm{Var}[L] + (\mathbb{E}[L] - \mu_w)^2}{2\sigma_w^2} = \tfrac{1}{2}\log(2\pi e\sigma_w^2).$$

The entropy of the log-normal is $H(L) = \mu_\ell + \frac{1}{2}\log(2\pi e\sigma_\ell^2)$. Therefore

$$\mathrm{KL}(L\|\mathcal{N}(\mu_w, \sigma_w^2)) = \tfrac{1}{2}\log(2\pi e\sigma_w^2) - \mu_\ell - \tfrac{1}{2}\log(2\pi e\sigma_\ell^2) = \tfrac{1}{2}\log\tfrac{\sigma_w^2}{\sigma_\ell^2} - \mu_\ell. \tag{26}$$

Substitute $\sigma_\ell^2 = \log(1 + 1/r_w^2) = 1/r_w^2 - 1/(2r_w^4) + O(1/r_w^6)$, $\sigma_w^2 = \mu_w^2/r_w^2$, and $\mu_\ell = \log|\mu_w| - \sigma_\ell^2/2$ into (26). Expand $\log(\sigma_\ell^2) = \log(1/r_w^2(1 - 1/(2r_w^2) + O(1/r_w^4))) = -2\log r_w - 1/(2r_w^2) +$

$O(1/r_w^4)$, so

$$\begin{aligned}\tfrac{1}{2}\log\frac{\sigma_w^2}{\sigma_\ell^2} &= \tfrac{1}{2}\left[\log\frac{\mu_w^2}{r_w^2} - \log\sigma_\ell^2\right] \\ &= \tfrac{1}{2}\left[2\log|\mu_w| - 2\log r_w + 2\log r_w + \tfrac{1}{2r_w^2} + O(1/r_w^4)\right] \\ &= \log|\mu_w| + \frac{1}{4r_w^2} + O(1/r_w^4).\end{aligned}$$

Using $-\mu_\ell = -\log|\mu_w| + \sigma_\ell^2/2 = -\log|\mu_w| + 1/(2r_w^2) + O(1/r_w^4)$:

$$\mathrm{KL} = \underbrace{\log|\mu_w| + \frac{1}{4r_w^2}}_{\frac{1}{2}\log(\sigma_w^2/\sigma_\ell^2)} \underbrace{- \log|\mu_w| + \frac{1}{2r_w^2}}_{-\mu_\ell} + O(1/r_w^4) = \frac{3}{4r_w^2} + O(1/r_w^4).$$

The factor $\frac{3}{4}$ is the sum of two $O(r_w^{-2})$ contributions: $\frac{1}{4}$ from expanding $\log(\sigma_w^2/\sigma_\ell^2)$ and $\frac{1}{2}$ from $\sigma_\ell^2/2$; this is (25). □

Lemma B.4 (Moment error of the log-normal backward approximation)**.** *Let* $r = \min(|\mu_y|/\sigma_y, |\mu_z|/\sigma_z) \geq 2$. *Denote by* μ_X^* *and* $(\sigma_X^*)^2$ *the mean and variance of* $\tilde{m}_{f\to X}^r$ *(equation (24)), and by* $\hat{\mu}_X := \hat{\tau}_x/\hat{\rho}_x$ *and* $\hat{\sigma}_X^2 := 1/\hat{\rho}_x$ *the DMA moments from Proposition 3.2. Then*

$$|\hat{\mu}_X - \mu_X^*| = O\left(\frac{|\mu_z/\mu_y|}{r^2}\right), \qquad |\hat{\sigma}_X^2 - (\sigma_X^*)^2| = O\left(\frac{(\sigma_X^*)^2}{r^2}\right), \tag{27}$$

and consequently

$$\mathrm{KL}\left(N_X^* \,\middle\|\, \hat{m}_{f\to X}\right) = O\left(\frac{1}{r^2}\right), \tag{28}$$

where $N_X^* = \mathcal{N}(\mu_X^*, (\sigma_X^*)^2)$.

Proof sketch. All moments are conditional on $A_r = \{|Y| \geq |\mu_y|/2\}$; the unconditional ratio Z/Y has no finite moments. Without loss of generality, $\mu_y, \mu_z > 0$; write $Y = \mu_y(1+\eta U)$, $Z = \mu_z(1+\xi V)$, $U, V \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $\eta = 1/r_y$, $\xi = 1/r_z$, $a = \mu_z/\mu_y$.

Central-region Taylor expansion. Let $B_r = \{|U| \leq K_r\}$, $K_r = \sqrt{16\log r}$ (as in Lemma B.5), and restrict to $A_r^+ \cap B_r$ (the exponentially small branches A_r^- and $A_r \cap B_r^c$ are treated below). On $A_r^+ \cap B_r$, $|\eta U| \leq K_r/r_y \to 0$, so the Taylor expansion

$$\frac{1}{1+\eta U} = 1 - \eta U + (\eta U)^2 + R_3, \qquad |R_3| \leq 2|\eta U|^3 \leq 2(K_r/r_y)^3,$$

holds uniformly (the denominator satisfies $1+\eta U \geq 1/2$ on A_r^+). Hence

$$X = a(1+\xi V)(1 - \eta U + (\eta U)^2 + R_3).$$

Taking $\mathbb{E}[\cdot \mid A_r^+ \cap B_r]$ and using $\mathbb{E}[U^k \mid A_r^+] = \mathbb{E}[U^k] + O(e^{-cr^2})$ for $k \leq 4$ (since A_r^+ has probability $1 - O(e^{-cr^2})$):

$$\mu_X^* \;=\; a(1+\eta^2) + O_\kappa\left(a\frac{(\log r)^{3/2}}{r^3}\right) \;=\; a(1+\eta^2) + o_\kappa(ar^{-2}),$$

where the leading remainder comes from $\mathbb{E}[|\eta U|^3 \mid B_r] = O(\eta^3 K_r^3) = O((\log r)^{3/2}/r^3)$.

Tail-region bound. On $A_r \cap B_r^c$, $|1+\eta U| \geq 1/2$, so $|X| \leq 2|a|(1+|\xi V|)$. Gaussian tail bounds give $P(|U| > K_r) = O(r^{-4})$, so $\mathbb{E}[|X|\,\mathbf{1}_{B_r^c} \mid A_r] = O(|a|\, r^{-4} \cdot \mathrm{poly}(r)) = o(r^{-2})$. The branch A_r^- has probability $\Phi(-3r_y/2) = O(e^{-9r_y^2/8})$, contributing negligibly.

Comparison with the DMA moments. Let $D = \eta^2 + \xi^2$. The DMA mean $\hat{\mu}_X = a\, e^{D/2}$ (Lemma B.1) expands to $a(1 + D/2 + O(r^{-4})) = a(1 + \eta^2/2 + \xi^2/2 + O(r^{-4}))$. Combined with $\mu_X^* = a(1+\eta^2) + o_\kappa(ar^{-2})$:

$$\hat{\mu}_X - \mu_X^* = a\left(\tfrac{\xi^2 - \eta^2}{2}\right) + o_\kappa(ar^{-2}) = O_\kappa\left(\frac{|\mu_z/\mu_y|}{r^2}\right).$$

Variance error. We bound $(\sigma_X^*)^2 = \mathbb{E}[X^2 \mid A_r] - (\mu_X^*)^2$ directly. For the second moment, expand one order further: $(1+t)^{-2} = 1 - 2t + 3t^2 - 4t^3 + R_4(t)$ with $|R_4(t)| \le C|t|^4$ and $t = \eta U$. Since $\sup_{B_r} |\eta U| = K_r/r_y = o(1)$, this expansion is uniform on $A_r^+ \cap B_r$. Using $\mathbb{E}[U] = 0$, $\mathbb{E}[U^2] = 1$, $\mathbb{E}[U^3] = 0$ (odd moments of $\mathcal{N}(0,1)$ vanish; conditioning on A_r^+ changes these by $O(e^{-cr^2})$), and $\mathbb{E}[U^4] = 3$:

$$\mathbb{E}\left[\frac{1}{(1+\eta U)^2} \,\middle|\, A_r\right] = 1 + 3\eta^2 + O_\kappa(\eta^4) = 1 + 3\eta^2 + O_\kappa(r^{-4}),$$

where the $-4\eta^3\mathbb{E}[U^3]$ term vanishes, the R_4 remainder contributes $O(\eta^4\mathbb{E}[U^4\mathbf{1}_{B_r}]) = O(\eta^4) = O(r^{-4})$, and the contributions of B_r^c and A_r^- are $O_\kappa(r^{-4})$ after multiplying by their exponentially small probabilities. Since $\mathbb{E}[X^2 \mid A_r] = a^2(1+\xi^2)\,\mathbb{E}[(1+\eta U)^{-2} \mid A_r]$:

$$\mathbb{E}[X^2 \mid A_r] = a^2(1+\xi^2)(1 + 3\eta^2 + O_\kappa(D/r^2)).$$

Together with $(\mu_X^*)^2 = a^2(1+\eta^2)^2 + o_\kappa(a^2 r^{-2}) = a^2(1 + 2\eta^2 + O_\kappa(D/r^2))$:

$$(\sigma_X^*)^2 = a^2(\eta^2 + \xi^2) + O_\kappa(a^2 D/r^2) = a^2 D\big(1 + O_\kappa(r^{-2})\big).$$

The DMA variance $\hat\sigma_X^2 = (e^D - 1)\hat\mu_X^2 = a^2 D e^{2D}(1 + O(D)) = a^2 D(1 + O_\kappa(r^{-2}))$, so

$$\frac{|\hat\sigma_X^2 - (\sigma_X^*)^2|}{(\sigma_X^*)^2} = O_\kappa\left(\frac{1}{r^2}\right).$$

For (28), substitute into the exact Gaussian KL:

$$\mathrm{KL}(N_X^* \| \hat m_{f\to X}) = \frac{(\hat\mu_X - \mu_X^*)^2}{2\hat\sigma_X^2} + \frac{(\sigma_X^*)^2}{2\hat\sigma_X^2} - \frac{1}{2} - \frac{1}{2}\log\frac{(\sigma_X^*)^2}{\hat\sigma_X^2} = O\left(\frac{1}{r^2}\right).$$

□

Lemma B.5 (Non-Gaussianity of the truncated ratio)**.** *Assume additionally that the ratio of the two signal-to-noise ratios is bounded: $\kappa^{-1} \le r_y/r_z \le \kappa$ for some fixed $\kappa \ge 1$. Then there exists a constant $C = C(\kappa)$ such that, for all sufficiently large r (depending only on κ),*

$$\mathrm{KL}\Big(\tilde m_{f\to X}^r \,\Big\|\, N_X^*\Big) \;\le\; \frac{C}{r^2}. \tag{29}$$

In particular, $\mathrm{KL}(\tilde m_{f\to X}^r \| N_X^*) = O_\kappa(r^{-2})$ *as* $r \to \infty$.

Proof. By changing signs if necessary, assume $\mu_y > 0$ and $\mu_z > 0$. Write $Y = \mu_y(1+\eta U)$ and $Z = \mu_z(1+\xi V)$ with $U, V \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, $\eta = 1/r_y$, $\xi = 1/r_z$. Set $a = \mu_z/\mu_y$, $D = \eta^2 + \xi^2$, $\rho = -\eta/\sqrt{D}$, $\tau = \xi/\sqrt{D}$, so $\rho^2 + \tau^2 = 1$ and $|\rho|, |\tau| \asymp_\kappa 1$. Let $W = (X - \mu_X^*)/s_X$ be the standardised version of X under P_r, so N_X^* is exactly $\mathcal{N}(0,1)$ in W-coordinates.

Step 1: Conditional Gaussianity. With $\mu_y > 0$, the truncation event is $A_r = \{|Y| \ge \mu_y/2\} = A_r^+ \cup A_r^-$ where $A_r^+ = \{1 + \eta U \ge 1/2\}$ and $A_r^- = \{1 + \eta U \le -1/2\}$. The negative branch satisfies $\mathbb{P}(A_r^-) = \Phi(-3r_y/2) \le e^{-9r_y^2/8}$, which is exponentially small; its contribution to all subsequent expectations is absorbed into the $O(e^{-cr^2})$ remainder and we work hereafter on A_r^+. On A_r^+ we have $1 + \eta u \ge 1/2 > 0$, so the denominator is bounded away from zero, and for fixed $U = u$, $X = a(1+\xi V)/(1+\eta u)$ is a linear function of the Gaussian V. Hence $P_{W|U=u,A_r^+}$ is exactly $\mathcal{N}(m_u, v_u)$ with

$$m_u = \frac{a/(1+\eta u) - \mu_X^*}{s_X}, \qquad v_u = \frac{a^2\xi^2}{s_X^2(1+\eta u)^2}.$$

Step 2: Per-slice KL bound. Let $B_r = \{|U| \le K_r\}$ with $K_r = \sqrt{16\log r}$. For $u \in B_r$, a Taylor expansion of $(1+\eta u)^{-1}$ together with $\mu_X^* = a(1+\eta^2) + o_\kappa(ar^{-2})$ (Lemma B.4) and $s_X \asymp_\kappa a\sqrt{D}$ gives

$$m_u - \rho u = O_\kappa\left(\frac{1+u^2}{r}\right), \qquad v_u - \tau^2 = O_\kappa\left(\frac{1+|u|}{r}\right).$$

Since τ is bounded away from zero, the Gaussian KL formula yields

$$\mathrm{KL}\Big(\mathcal{N}(m_u, v_u) \,\Big\|\, \mathcal{N}(\rho u, \tau^2)\Big) \le \frac{C_\kappa(1+u^4)}{r^2}, \qquad u \in B_r.$$

On $B_r^c \cap A_r$ the KL is at most $C_\kappa(1+r^2+u^2)$; Gaussian tail bounds give $\mathbb{P}(|U| > K_r \mid A_r) \le Cr^{-4}$, so $\mathbb{E}[(1+r^2+U^2)\mathbf{1}_{B_r^c} \mid A_r] \le C_\kappa r^{-2}$.

Step 3: Chain rule and data-processing. Define the reference joint $Q_{W,U}(\mathrm{d}w, \mathrm{d}u) = \phi(u)\,\mathcal{N}(w; \rho u, \tau^2)\,\mathrm{d}w\,\mathrm{d}u$. The KL chain rule gives

$$\mathrm{KL}(P_{W,U} \| Q_{W,U}) = \underbrace{\mathrm{KL}(P_{U|A_r} \| \mathcal{N}(0,1))}_{=-\log \mathbb{P}(A_r) = O(e^{-cr^2})} + \mathbb{E}_{U|A_r}\left[\mathrm{KL}(P_{W|U,A_r} \| \mathcal{N}(\rho U, \tau^2))\right] \le \frac{C_\kappa}{r^2}.$$

The marginal of $Q_{W,U}$ in W is $\mathcal{N}(0,1)$ (since $W_0 = \rho U + \tau V \sim \mathcal{N}(0, \rho^2+\tau^2) = \mathcal{N}(0,1)$). The data-processing inequality applied to $(W,U) \mapsto W$ gives

$$\mathrm{KL}(P_r \| N_X^*) = \mathrm{KL}(P_W \| \mathcal{N}(0,1)) \le \mathrm{KL}(P_{W,U} \| Q_{W,U}) \le \frac{C_\kappa}{r^2}. \qquad \square$$

Proof of Theorem 3.3. Let $N_X^* = \mathcal{N}(\mu_X^*, (\sigma_X^*)^2)$ be the moment-matched Gaussian of $\tilde{m}_{f\to X}^r$. Since the log-density ratio of any two Gaussians is a quadratic polynomial, and $\tilde{m}_{f\to X}^r$ and N_X^* share the same first two moments:

$$\mathbb{E}_{\tilde{m}^r}\left[\log \frac{N_X^*(X)}{\hat{m}_{f\to X}(X)}\right] = \mathbb{E}_{N_X^*}\left[\log \frac{N_X^*(X)}{\hat{m}_{f\to X}(X)}\right].$$

This gives the exact Pythagorean decomposition

$$\mathrm{KL}\Big(\tilde{m}_{f\to X}^r \,\Big\|\, \hat{m}_{f\to X}\Big) = \underbrace{\mathrm{KL}\Big(\tilde{m}_{f\to X}^r \,\Big\|\, N_X^*\Big)}_{\le C_\kappa/r^2 \text{ (Lemma B.5)}} + \underbrace{\mathrm{KL}\Big(N_X^* \,\Big\|\, \hat{m}_{f\to X}\Big)}_{O_\kappa(1/r^2) \text{ (Lemma B.4)}}. \tag{30}$$

Both terms are $O_\kappa(1/r^2)$ as $r \to \infty$, giving (8). $\square$

B.6 Proofs of Propositions 3.4 and 3.5 (ReLU Factor DMA)

Remark B.6 (Improper Backward Message at $\alpha = 0$)**.** *For the standard ReLU ($\alpha = 0$), every $x \le 0$ maps to $y = 0$, so the $x \le 0$ piece of the backward message evaluates to the constant $\mathcal{N}(0;\, \mu_y, \sigma_y^2) = \varphi(v)/\sigma_y$ rather than a Gaussian in x. This constant piece has infinite mass, making the backward message improper. The formulas (11)–(12) are inapplicable at $\alpha = 0$; in practice, one uses $\alpha = \epsilon \ll 1$ (leaky ReLU) or replaces the backward message with a uniform prior on the negative half-line. For $\mu_y \gg \sigma_y$ the constant is negligible and the backward message is approximately $\mathcal{N}(\mu_y, \sigma_y^2)$, but no finite normalisation correction applies in general.*

Truncated Gaussian lemma. Both proofs use the following standard integral identities. For $X \sim \mathcal{N}(\mu, \sigma^2)$, $u = \mu/\sigma$, $P = \Phi(u)$, $\phi = \varphi(u)$:

$$\int_{-\infty}^{0} \mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = 1 - P, \qquad \int_{0}^{\infty} \mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = P, \tag{31}$$

$$\int_{-\infty}^{0} x\,\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = \mu(1-P) - \sigma\phi, \qquad \int_{0}^{\infty} x\,\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = \mu P + \sigma\phi, \tag{32}$$

$$\int_{-\infty}^{0} x^2\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = (\mu^2+\sigma^2)(1-P) - \mu\sigma\phi, \quad \int_{0}^{\infty} x^2\mathcal{N}(x;\mu,\sigma^2)\,\mathrm{d}x = (\mu^2+\sigma^2)P + \mu\sigma\phi. \tag{33}$$

These follow from completing the square in the exponent and the Gaussian survival function identity $\mathbb{E}[X\mathbf{1}_{X>0}] = \mu\Phi(\mu/\sigma) + \sigma\varphi(\mu/\sigma)$.

Proof of Proposition 3.4. The exact forward message is the pushforward of $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ through $\mathrm{ReLU}_\alpha(\cdot)$. We compute $\mathbb{E}[Y] = \mathbb{E}[\mathrm{ReLU}_\alpha(X)]$ and $\mathbb{E}[Y^2] = \mathbb{E}[\mathrm{ReLU}_\alpha(X)^2]$ using (33) with $u = \mu_x/\sigma_x$, $P = \Phi(u)$, $\phi = \varphi(u)$.

For the first moment, writing $\mathrm{ReLU}_\alpha(x) = x\mathbf{1}_{x>0} + \alpha x\mathbf{1}_{x\le 0}$:

$$\begin{aligned}\mathbb{E}[Y] &= \int_0^\infty x\mathcal{N}(x;\mu_x,\sigma_x^2)\,\mathrm{d}x + \alpha\int_{-\infty}^0 x\mathcal{N}(x;\mu_x,\sigma_x^2)\,\mathrm{d}x \\ &= (\mu_x P + \sigma_x\phi) + \alpha(\mu_x(1-P) - \sigma_x\phi) \\ &= \mu_x[\alpha + (1-\alpha)P] + (1-\alpha)\sigma_x\phi = \mu_x A + (1-\alpha)\sigma_x\phi.\checkmark\end{aligned}$$

For the second moment:

$$\begin{aligned}\mathbb{E}[Y^2] &= \int_0^\infty x^2\mathcal{N}(x;\mu_x,\sigma_x^2)\,\mathrm{d}x + \alpha^2\int_{-\infty}^0 x^2\mathcal{N}(x;\mu_x,\sigma_x^2)\,\mathrm{d}x \\ &= [(\mu_x^2+\sigma_x^2)P + \mu_x\sigma_x\phi] + \alpha^2[(\mu_x^2+\sigma_x^2)(1-P) - \mu_x\sigma_x\phi] \\ &= (\mu_x^2+\sigma_x^2)[\alpha^2 + (1-\alpha^2)P] + (1-\alpha^2)\mu_x\sigma_x\phi = (\mu_x^2+\sigma_x^2)B + (1-\alpha^2)\mu_x\sigma_x\phi.\checkmark\end{aligned}$$

The DMA is the moment-matching projection $\hat{m}_{f\to Y} = \mathcal{N}(m_Y, s_Y^2)$ with $m_Y = \mathbb{E}[Y]$ and $s_Y^2 = \mathbb{E}[Y^2] - m_Y^2$.

Consistency. As $\sigma_x \to 0$ with $\mu_x \to \bar{x}$: $u \to \pm\infty$, so for $\bar{x} > 0$: $P \to 1$, $\phi \to 0$, $A \to 1$, $B \to 1$, giving $m_Y \to \bar{x}$ and $s_Y^2 \to 0$; for $\bar{x} < 0$: $P \to 0$, $A \to \alpha$, $B \to \alpha^2$, giving $m_Y \to \alpha\bar{x}$ and $s_Y^2 \to 0$. Both recover $\mathrm{ReLU}_\alpha(\bar{x})$ exactly. □

Proof of Proposition 3.5. The exact backward message density is

$$m_{f\to X}(x) = \begin{cases}\mathcal{N}(x;\,\mu_y,\sigma_y^2) & x > 0,\\ \mathcal{N}(\alpha x;\,\mu_y,\sigma_y^2) & x \le 0,\end{cases} \tag{34}$$

obtained by integrating the factor $\delta(y - \mathrm{ReLU}_\alpha(x))$ against $m_{Y\to f}(y) = \mathcal{N}(y;\mu_y,\sigma_y^2)$. (For $x > 0$, the constraint $y = x$ gives $m_{f\to X}(x) = \mathcal{N}(x;\mu_y,\sigma_y^2)$; for $x \le 0$, the constraint $y = \alpha x$ gives $m_{f\to X}(x) = \mathcal{N}(\alpha x;\mu_y,\sigma_y^2)$.)

Total mass. Setting $v = \mu_y/\sigma_y$, $P = \Phi(v)$, $Q = 1 - P$:

$$\int_{-\infty}^{\infty} m_{f\to X}(x)\,\mathrm{d}x = P + \frac{1}{\alpha}\int_{-\infty}^{0} \mathcal{N}(t;\mu_y,\sigma_y^2)\,\mathrm{d}t = P + \frac{Q}{\alpha} = \frac{\tilde{C}}{\alpha},$$

where the substitution $t = \alpha x$ (for $\alpha > 0$) contributes the $1/\alpha$ factor and $\tilde{C} = \alpha P + Q$.

First moment. Using (33) and the substitution $t = \alpha x$ for the $x \le 0$ piece:

$$\begin{aligned}\int_{-\infty}^{\infty} x\, m_{f\to X}(x)\,\mathrm{d}x &= \int_{0}^{\infty} x\mathcal{N}(x;\mu_y,\sigma_y^2)\,\mathrm{d}x + \int_{-\infty}^{0} x\mathcal{N}(\alpha x;\mu_y,\sigma_y^2)\,\mathrm{d}x \\ &= (\mu_y P + \sigma_y\phi) + \frac{1}{\alpha^2}(\mu_y Q - \sigma_y\phi).\end{aligned}$$

Dividing by $\tilde{C}/\alpha$:

$$m_X = \frac{\mu_y(\alpha^2 P + Q) + (\alpha^2 - 1)\sigma_y\phi}{\alpha\tilde{C}}.\checkmark$$

Second moment.

$$\int_{-\infty}^{\infty} x^2\, m_{f\to X}(x)\,\mathrm{d}x = (\mu_y^2 + \sigma_y^2)P + \mu_y\sigma_y\phi + \frac{1}{\alpha^3}[(\mu_y^2 + \sigma_y^2)Q - \mu_y\sigma_y\phi].$$

Dividing by $\tilde{C}/\alpha$:

$$\mathbb{E}[X^2] = \frac{(\mu_y^2 + \sigma_y^2)(\alpha^3 P + Q) + (\alpha^3 - 1)\mu_y\sigma_y\phi}{\alpha^2\tilde{C}},$$

giving $s_X^2 = \mathbb{E}[X^2] - m_X^2$ as stated.

Consistency. As $\sigma_y \to 0$ with $\mu_y \to \bar{y}$, $v \to \pm\infty$ and $\phi \to 0$. For $\bar{y} > 0$: $P \to 1$, $Q \to 0$, $\tilde{C} \to \alpha$, giving $m_X \to \bar{y}$ and $s_X^2 \to 0$. For $\bar{y} < 0$ (requires $\alpha > 0$): $P \to 0$, $Q \to 1$, $\tilde{C} \to 1$, giving $m_X \to \bar{y}/\alpha$ and $s_X^2 \to 0$. Both recover the exact inverse $x = \mathrm{ReLU}_\alpha(\bar{y})^{-1}$. □

B.7 Theorem Coverage by Factor and Direction

Table 2 summarises which theorem governs each factor direction in the BNN factor graph. Two tiers arise: *Tier 1* (proper messages) covered by Theorem 2.3; *Tier 2* (improper messages) covered by the concentrated-input Theorem 3.3.

Table 2: Theorem coverage for each factor direction in the BNN. "Proper?" refers to whether the DMA message is a valid (positive-precision) Gaussian. All linear and copy factors satisfy Corollary 2.4 exactly; the Gaussian likelihood is conjugate and exact. See Remarks B.2 and B.6 for the improper cases.

Factor	Direction	Proper?	Theorem
Gaussian prior	both	Yes	Cor. 2.4 (exact)
Gaussian likelihood	both	Yes	exact (conjugate)
Linear / copy	both	Yes	Cor. 2.4
Product ($z{=}xy$)	fwd (z)	Yes	Thm. 2.3
Product ($z{=}xy$)	bwd (x, y)	No	Thm. 3.3
Leaky-ReLU ($\alpha{>}0$)	fwd	Yes	Thm. 2.3
Leaky-ReLU ($\alpha{>}0$)	bwd	Yes	Thm. 2.3
Standard ReLU ($\alpha{=}0$)	fwd	Yes	Thm. 2.3
Standard ReLU ($\alpha{=}0$)	bwd	No	Rem. B.6 (no bound derived†)

†Standard ReLU ($\alpha{=}0$) is not used in this paper; a concentrated-input bound analogous to Thm. 3.3 is left for future work.

C Empirical Validation of DMA Approximation Quality

This appendix provides six layers of empirical validation:

1. **Master theorem** (§C.1): we directly verify Theorem 2.3 by computing the message KL δ and the resulting marginal KL for 192 leaky-ReLU factor configurations and confirming that the bound $\|m_{X_j \to f}\|_\infty / Z \cdot \delta$ holds in every case.
2. **Bound tightness** (§C.2): we sweep each parameter axis individually (input SNR r, leaky slope α, input width σ_x) and overlay actual marginal KL with the theoretical bound, showing how the gap varies across the full parameter range.
3. **Factor-level SNR sweep** (§C.3): we verify the end-to-end $O(1/r^2)$ rate for the product backward message (Theorem 3.3) and confirm the same empirical rate for the leaky-ReLU backward message via importance-sampling reference marginals over a range of input SNR values.
4. **Copy factor** (§C.4): we visualise the $O(1/r^2)$ Gaussian–log-normal KL bound from Lemma B.3 as a 2D heatmap over the parameter space, confirming both horizontal bands (left) and diagonal r-contours (right).
5. **Product factor** (§C.5): we compare DMA marginals against IS references for nominal and stress configurations, illustrating where the Gaussian approximation holds and where it degrades.
6. **ReLU factor** (§C.6): we compare DMA forward and backward marginals against IS references for the leaky-ReLU factor, showing the approximation quality in both directions.

C.1 Master Theorem Validation

We validate Theorem 2.3 directly: for 192 leaky-ReLU factor configurations spanning $\alpha \in \{0.1, 0.3, 0.5\}$, incoming-message widths $\sigma_x \in \{0.5, 1.0, 1.5, 2.0\}$, and factor SNR values $r = |\mu_y|/\sigma_y \in [0.5, 7]$, we compute both

$$\delta = \mathrm{KL}(m_{f \to X} \,\|\, \hat{m}_{f \to X})$$

and the resulting marginal error $\mathrm{KL}(p_X/Z \,\|\, \hat{p}_X/\hat{Z})$ analytically on a fine grid (no Monte Carlo). The true backward message $m_{f \to X}(x) = \mathcal{N}(\mathrm{relu}(x; \alpha);\, \mu_y, \sigma_y^2)$ is a proper two-piece Gaussian for all $\alpha > 0$, so it lies within the scope of Theorem 2.3. Figure 3 shows the results.

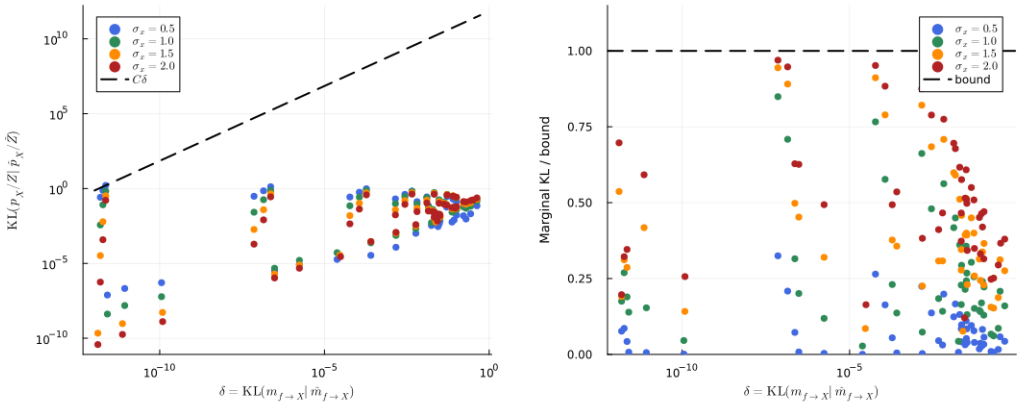


Figure 3: Empirical validation of Theorem 2.3 across 192 leaky-ReLU factor configurations. *Left*: log-log scatter of message error δ versus marginal error; the dashed line $C\delta$ (with $C = \|m_{X_j \to f}\|_\infty / Z$) lies above every point, confirming $O(\delta)$ scaling. *Right*: the marginal KL normalised by its per-configuration bound $\|m_{X_j \to f}\|_\infty / Z \cdot \delta$; all 192 ratios lie below the bound line at 1 (maximum ratio 0.97), directly certifying the theorem. The bound is tight: the $\|m_{X_j \to f}\|_\infty / Z$ prefactor evaluates the Gaussian sup-norm, which is largest where the incoming message is most concentrated and the marginal error is therefore small anyway.

C.2 Bound Tightness: Per-Parameter Sweeps

Figure 4 isolates each parameter axis in turn, holding the others fixed, and overlays the actual marginal KL (solid) with the theoretical bound (dashed) to show directly how the gap varies. Three qualitatively different behaviors emerge. *Panel (a) — SNR r*: the bound decreases monotonically, while the actual marginal KL is non-monotone: small at low r (where the DMA moment-match of a near-symmetric message is accurate), peaking at moderate $r \approx 2$–3 (where the asymmetric kink of the leaky ReLU is hardest to capture), then declining again as both messages concentrate at high r. *Panel (b) — leaky slope α*: both quantities decrease monotonically as $\alpha \to 1$ (the factor approaches a linear copy, which is conjugate and has $\delta = 0$) and grow as $\alpha \to 0$ (approaching the improper hard-ReLU limit). *Panel (c) — incoming width σ_x*: widening the incoming message at fixed r lets more backward-message approximation error propagate into the marginal, so the actual KL increases; the bound's prefactor $\|m_{X_j \to f}\|_\infty / Z \propto 1/\sigma_x$ simultaneously decreases (the Gaussian sup-norm decays as $1/\sigma_x$ while Z grows more slowly). The bound is therefore most conservative at small σ_x, where the narrow incoming message masks the backward approximation error in the marginal.

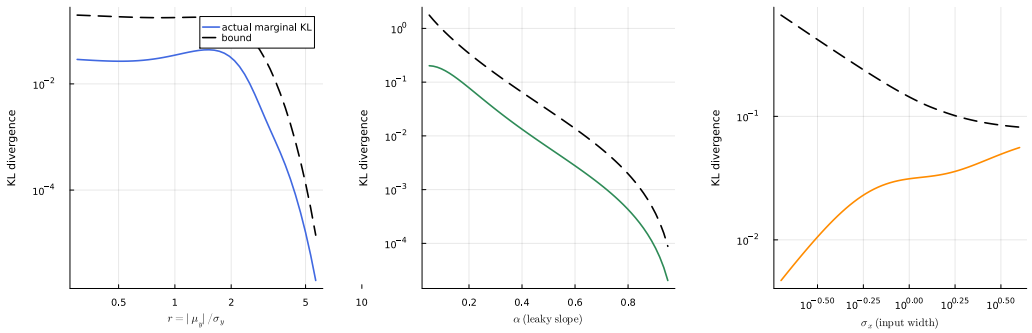


Figure 4: Actual marginal KL (solid, coloured) versus the theoretical bound $\|m_{X_j \to f}\|_\infty / Z \cdot \delta$ (dashed, black) as each parameter is swept individually (legend shown in panel (a)). *(a)* Input SNR r ($\alpha = 0.3$, $\sigma_x = 1$ fixed): the bound decays monotonically; the actual KL is non-monotone, peaking near $r \approx 2$–3 where the ReLU kink is hardest to match, and small at both low and high r. *(b)* Leaky slope α ($r = 2$, $\sigma_x = 1$ fixed): both decrease as $\alpha \to 1$ (linear factor, $\delta = 0$) and grow as $\alpha \to 0$ (approaching the improper hard-ReLU limit). *(c)* Input width σ_x ($r = 2$, $\alpha = 0.3$ fixed): the actual KL increases with σ_x (wider incoming message exposes more backward-message error in the marginal), while the bound decreases through its $1/\sigma_x$ prefactor; the conservatism is greatest at small σ_x. In all panels the bound lies strictly above the actual KL, with a maximum ratio of 0.68 across all three sweeps (width sweep, panel (c)).

C.3 SNR Sweep: Factor-Level Approximation Quality

Setup. For each factor and each SNR value r, we draw $N = 500{,}000$ samples from the incoming Gaussian messages, propagate them through the factor's deterministic relation, and weight them by the remaining incoming message to obtain an IS estimate of the true marginal. We then compute $\mathrm{KL}(p_{\mathrm{IS}} \| q_{\mathrm{DMA}})$ via a normalized weighted histogram with 500 bins covering the 0.1%–99.9% quantile range of the IS distribution. The dashed reference line C/r^2 is fitted to the high-SNR region ($r \geq 4$) by taking the median of $\mathrm{KL}(r) \cdot r^2$. Figure 5 shows the results for both factors.

C.4 Copy Factor: Gaussian–Log-Normal Approximation

The product backward message (Proposition 3.2) is derived via two *copy-factor* steps that bridge the Gaussian and log-normal families. *Step 1* (backward): the exact backward message $\mathrm{LN}(\cdot;\, \mu_Y, \sigma_Y^2)$ (log-space parameterised) is approximated by a moment-matched Gaussian $\hat{m}_{f \to X}$. *Step 2* (forward): the Gaussian message $\mathcal{N}(\cdot;\, \mu_X, \sigma_X^2)$ is approximated by a moment-matched log-normal LN_{m}. Both introduce $O(1/r^2)$ KL error (Lemma B.3); Figure 6 confirms this empirically.

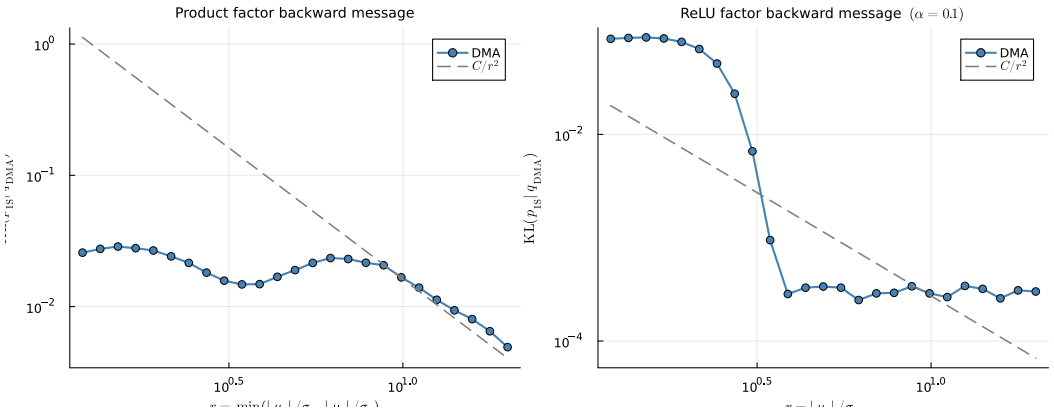


Figure 5: KL divergence between IS reference marginal and DMA marginal as a function of input SNR r. *Left*: product factor backward message to X from $Z = XY$; incoming messages $Y \sim \mathcal{N}(\mu_y, (\mu_y/r)^2)$ and $Z \sim \mathcal{N}(\mu_z, (\mu_z/r)^2)$ with $\mu_y = 4$, $\mu_z = 10$, and the incoming message from X fixed at $\mathcal{N}(3, 1)$. *Right*: ReLU backward message to X from $Y = \mathrm{relu}(X;\, \alpha)$ with $\alpha = 0.1$; incoming message $X \sim \mathcal{N}(\mu_x, (\mu_x/r)^2)$ with $\mu_x = 1$. Both panels show empirical $O(1/r^2)$ decay. The product backward (left) is covered by Theorem 3.3; the leaky-ReLU backward (right, α=0.1, proper message) is consistent with Theorem 2.3 applied to the message KL.

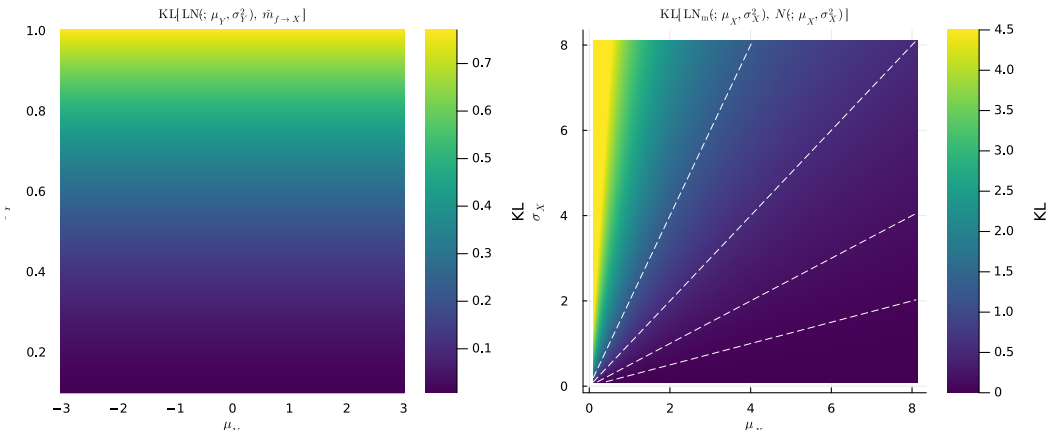


Figure 6: Copy-factor KL divergence over the 2D parameter space, computed from the closed-form expression in Lemma B.3 (no Monte Carlo). *Left*: Step 1 error $\mathrm{KL}[\mathrm{LN}(\cdot;\, \mu_Y, \sigma_Y^2),\ \hat{m}_{f\to X}(\cdot)]$ swept over (μ_Y, σ_Y). Horizontal bands confirm that the KL depends only on the log-space variance σ_Y^2, not on μ_Y; the analytical rate is $\frac{3}{4}\sigma_Y^2 + O(\sigma_Y^4)$. *Right*: Step 2 error $\mathrm{KL}[\mathrm{LN}_{\mathrm{m}}(\cdot;\, \mu_X, \sigma_X^2),\ \mathcal{N}(\cdot;\, \mu_X, \sigma_X^2)]$ swept over (μ_X, σ_X). Dashed white lines are level curves of $r = \mu_X/\sigma_X$; diagonal banding confirms the KL depends only on r, consistent with the $O(1/r^2)$ rate.

C.5 Product Factor

The left panel of Figure 5 shows the product backward message (to X given $Z = XY$). This is the case covered by Theorem 3.3: the exact message is improper (non-normalisable), so the IS reference is the true marginal of X rather than the message itself. The theorem predicts $O(1/r^2)$ total KL error; the sweep confirms this rate empirically across two decades of r. At $r = 2$ (the theorem's stated threshold for properness of the truncated reference), the KL is already small; at $r = 10$ it is several orders of magnitude smaller, tracking the dashed C/r^2 reference closely. Figure 7 compares DMA and IS marginals at a nominal and a stress configuration; the forward marginal (Z) is well approximated in both cases, while the backward marginals (X, Y) become increasingly non-Gaussian under stress, illustrating the regime where the $O(1/r^2)$ bound is most relevant.

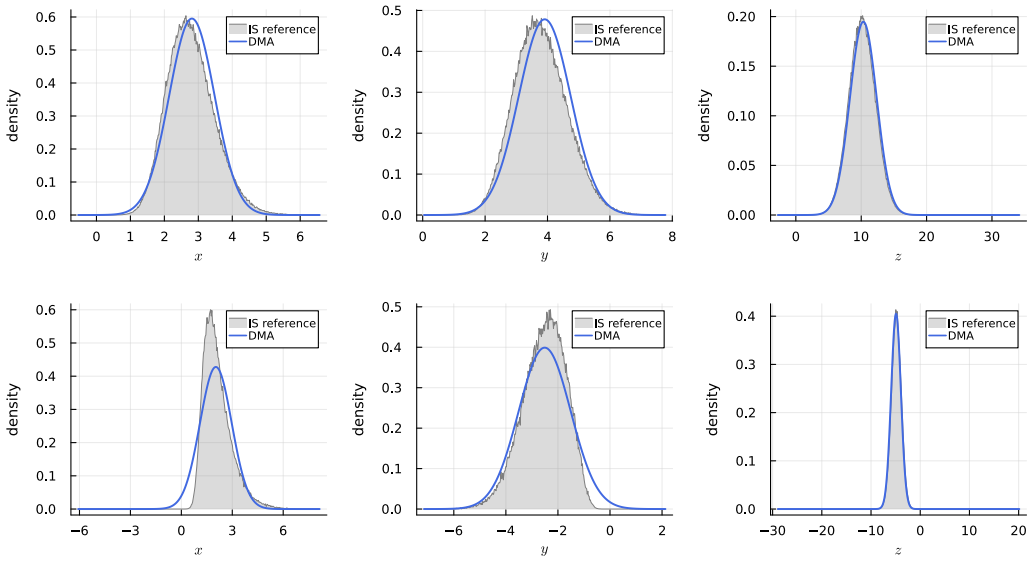


Figure 7: Product factor $f(X, Y, Z) = \delta(Z - XY)$. Each row shows marginals X, Y, Z (left to right). Blue: DMA. Gray: IS. (a) Nominal ($\mu_X = 3$, $\sigma_X^2 = 1$, $\mu_Y = 4$, $\sigma_Y^2 = 1$, $\mu_Z = 10$, $\sigma_Z^2 = 5$): the Z forward marginal is well approximated by the DMA Gaussian; the X and Y backward marginals are already noticeably skewed in the IS reference. (b) Stress ($\mu_X = 1$, $\sigma_X^2 = 4$, $\mu_Y = -2.5$, $\sigma_Y^2 = 1$, $\mu_Z = -5$, $\sigma_Z^2 = 1$): Z remains well approximated; the X and Y backward marginals become severely non-Gaussian.

C.6 ReLU Factor

The right panel of Figure 5 shows the leaky ReLU backward message ($\alpha = 0.1$). Unlike the product factor, the leaky ReLU backward message is always proper (the normalisation correction via the Mills ratio is finite for all $\alpha > 0$), so this case is covered directly by the master theorem. The empirical $O(1/r^2)$ decay is consistent with the general O(δ) bound, and the absolute KL values are lower than the product factor at matched r, reflecting the smoother shape of the leaky ReLU factor compared to the ratio $X = Z/Y$. The standard ReLU ($\alpha = 0$) backward message is improper; that case is analogous to the product factor and is not shown. Figure 8 shows DMA versus IS at a nominal input: the backward marginal IS reference is non-Gaussian (sharp, asymmetric), and the forward marginal IS has a point mass at $y = 0$ from the leaky branch; the DMA Gaussian captures the bulk location but cannot represent these non-Gaussian features.

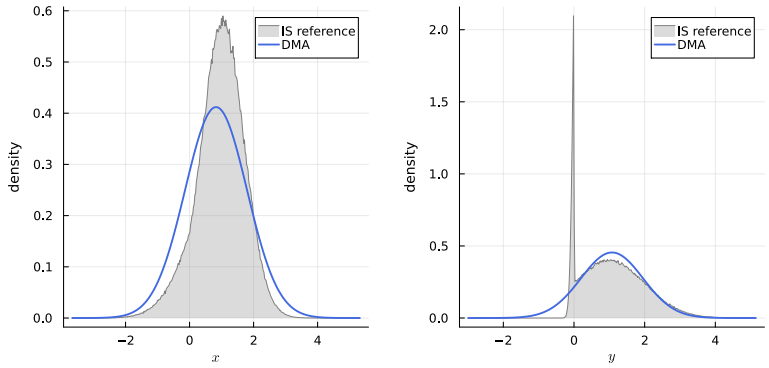


Figure 8: ReLU / Leaky-ReLU factor ($\alpha = 0.1$), nominal scenario ($\mu_X = 1$, $\sigma_X^2 = 1$). Left: backward marginal X. Right: forward marginal Y. Blue: DMA, Gray: IS reference

D INFERENCE ALGORITHM

Algorithm 1 gives the complete DMA BNN training procedure. We write $(\mu_{ij}^{(l)}, \sigma_{ij}^{2,(l)})$ for the mean and variance of $q_{W_{ij}^{(l)}}$.

Scalar decomposition. The matrix-vector product $z_i^{(l)} = \sum_{j=1}^{d_{l-1}} W_{ij}^{(l)} x_j^{(l-1)}$ decomposes into d_{l-1} product factors $z_{ij}^{(l)} = W_{ij}^{(l)} x_j^{(l-1)}$ and one linear sum factor per output unit i. Forward messages are summed exactly under independence (means and variances add). The backward message to $z_{ij}^{(l)}$ subtracts the total forward mean $\mu_{z_i}^{(l)}$ and re-adds $\mu_{z_{ij}}^{(l)}$; the backward variance adds $\sigma_{z_i}^{2,(l)}$ and subtracts $\sigma_{z_{ij}}^{2,(l)}$ (Gaussian deconvolution: exact for linear factors). The backward message to $x_j^{(l-1)}$ is accumulated in precision form (τ, ρ) over all output units i before conversion to moments.

Mini-batch EP structure. The N training examples are split into B mini-batches of size N/B. Each mini-batch b has one stored outgoing weight message $m_{l,ij}^{(b)}$, representing the combined likelihood contribution of that batch. Before processing batch b, its previous message is divided out of the current weight marginal to form the *incoming belief* $q_{W_{ij}^{(l)}}^{-b}$ (the product of the prior and all other batches' messages). After processing all examples in the batch, the new outgoing message is extracted as $m_{\text{new}}^{(b)} = q_W / q_W^{-b}$. Setting $B = N$ (one example per batch) recovers per-example EP.

Prior, likelihood, and convergence. Prior factors $\mathcal{N}(\mu_{ij}^{(l)}, \sigma_0^2)$ with He-style random means $\mu_{ij}^{(l)} \sim \mathcal{N}(0, 1/d_{l-1})$, the Gaussian likelihood $\mathcal{N}(y_n; z^{(L)}, \beta^2)$, and the optional activation prior factors $\mathcal{N}(0, \sigma_{\text{act},l}^2)$ on each pre-activation $z_i^{(l)}$ all contribute exact messages ($\delta = 0$); in the paper experiments all $\sigma_{\text{act},l}^2 = \infty$ (uniform, no prior). The observed input x_n enters as a point mass ($\sigma^2 = 0$). Convergence is checked after each full epoch via the normalised average log-likelihood $\mathcal{L}_e = \frac{1}{N} \sum_n \log p(y_n \mid x_n)$: training stops when $|\mathcal{L}_e - \mathcal{L}_{e-1}| / \max(\epsilon_{\text{floor}}, |\mathcal{L}_{e-1}|) < \varepsilon$, where ϵ_{floor} is a small numerical constant. Per-example cost: $O(\sum_l d_l \, d_{l-1})$.

Remark D.1. *On the first pass every $m^{(b)}$ is uniform (zero precision), so the initial division leaves the marginal unchanged. After convergence the weight marginal satisfies $q_{W_{ij}^{(l)}} \propto \mathcal{N}(\mu_{ij}^{(l)}, \sigma_0^2) \cdot \prod_{b=1}^{B} m_{l,ij}^{(b)}$: the prior message times one stored factor message per mini-batch. The forward pass uses the full current belief $q_{W_{ij}^{(l)}}$ (not a cavity), so within a batch each example conditions on the beliefs accumulated from the prior and all other batches without requiring a per-example cavity state. The convergence threshold ε controls only the outer epoch loop; it does not appear in any per-factor message computation and is not a learning rate.*

Algorithm 1: DMA BNN Training (1/2: initialisation and forward sweep)

Input: Data $\{(x_n, y_n)\}_{n=1}^{N}$; widths $d_0, \ldots, d_L$; leaky-ReLU slopes $\alpha_1, \ldots, \alpha_{L-1}$; mini-batches B; σ_0^2 (prior var.); β^2 (noise var.); $\sigma_{\text{act},l}^2$, $l = 1, \ldots, L$ (act. prior var.; ∞ = no prior); ε (conv. threshold); E (max epochs).

Output: Weight beliefs $\{q_{W_{ij}^{(l)}}\}_{l,i,j}$.

// $m_{l,ij}^{(b)}$: stored outgoing msg from mini-batch b to $W_{ij}^{(l)}$; all initialised to uniform

$\mu_{ij}^{(l)} \sim \mathcal{N}(0, 1/d_{l-1})$;

$q_{W_{ij}^{(l)}} \leftarrow \mathcal{N}(\mu_{ij}^{(l)}, \sigma_0^2)$ for all $l,\ i \le d_l,\ j \le d_{l-1}$

$m_{l,ij}^{(b)} \leftarrow$ uniform for all $b \le B,\ l,\ i \le d_l,\ j \le d_{l-1}$

for $e = 1, \ldots, E$ **do**

  for $b = 1, \ldots, B$ (mini-batch $\mathcal{B}_b$, $|\mathcal{B}_b| = N/B$) **do**

    $q_{W_{ij}^{(l)}}^{-b} \leftarrow q_{W_{ij}^{(l)}} \div m_{l,ij}^{(b)}$;

    $q_{W_{ij}^{(l)}} \leftarrow q_{W_{ij}^{(l)}}^{-b}$ for all $l,\ i \le d_l,\ j \le d_{l-1}$ // divide out old batch msg

    for *each* $n \in \mathcal{B}_b$ **do**

      $\mu_j^{(0)} \leftarrow x_{n,j}$,

      $\sigma_j^{2,(0)} \leftarrow 0$ for $j = 1, \ldots, d_0$ // point-mass input, $\delta = 0$

      for $l = 1$ **to** L **do**

        for $i = 1, \ldots, d_l$ **do**

          for $j = 1, \ldots, d_{l-1}$ **do**

            $\mu_{z_{ij}}^{(l)} \leftarrow \mu_{ij}^{(l)}\, \mu_j^{(l-1)}$ // product factor (Prop. 3.1)

            $\sigma_{z_{ij}}^{2,(l)} \leftarrow \sigma_{ij}^{2,(l)} (\mu_j^{(l-1)})^2 + \sigma_j^{2,(l-1)} (\mu_{ij}^{(l)})^2 + \sigma_{ij}^{2,(l)}\, \sigma_j^{2,(l-1)}$

          end

          $\mu_{z_i}^{(l)} \leftarrow \sum_j \mu_{z_{ij}}^{(l)}$;

          $\sigma_{z_i}^{2,(l)} \leftarrow \sum_j \sigma_{z_{ij}}^{2,(l)}$ // sum factor

          // Act. prior $\mathcal{N}(0, \sigma_{\text{act},l}^2)$: zero mean $\Rightarrow$ $\tau_{z_i}^{(l)}$ unchanged; $\sigma_{\text{act},l}^2 = \infty$ = no-op

          $\tau_{z_i}^{(l)} \leftarrow \mu_{z_i}^{(l)} / \sigma_{z_i}^{2,(l)}$;

          $\rho_{z_i}^{(l)} \leftarrow 1/\sigma_{z_i}^{2,(l)} + 1/\sigma_{\text{act},l}^2$ $\mu_{z_i}^{(l)} \leftarrow \tau_{z_i}^{(l)} / \rho_{z_i}^{(l)}$;

          $\sigma_{z_i}^{2,(l)} \leftarrow 1/\rho_{z_i}^{(l)}$

          if $l < L$ **then** $(\mu_i^{(l)},\ \sigma_i^{2,(l)}) \leftarrow \text{ReLUFwd}(\mu_{z_i}^{(l)}, \sigma_{z_i}^{2,(l)}, \alpha_l)$ // ReLU factor (Prop. 3.4)

        end

      end

      $(\mu_i^{\text{bwd},(L)},\ \sigma_i^{2,\text{bwd},(L)}) \leftarrow (y_{n,i},\ \beta^2)$ for $i = 1, \ldots, d_L$ // likelihood backward to $z^{(L)}$, $\delta = 0$

    end

    $m_{l,ij}^{(b)} \leftarrow q_{W_{ij}^{(l)}} \div q_{W_{ij}^{(l)}}^{-b}$ for all $l,\ i \le d_l,\ j \le d_{l-1}$ // extract new batch msg

  end

end

Algorithm 1: DMA BNN Training (2/2: backward sweep and convergence check)

// (per example $n \in \mathcal{B}_b$, mini-batch b, epoch e:)
// --- Backward sweep: directly update q_W (no per-example cavity) ---
for $l = L$ **to** 1 **do**
  $\tau_j^x \leftarrow 0$,
  $\rho_j^x \leftarrow 0$ for $j = 1, \dots, d_{l-1}$
  for $i = 1, \dots, d_l$ **do**
    for $j = 1, \dots, d_{l-1}$ **do**
      // Sum-factor backward to $z_{ij}^{(l)}$: Gaussian deconvolution (exact)
      $\mu_{ij}^{\text{cav}} \leftarrow \mu_i^{\text{bwd},(l)} - \mu_{z_i}^{(l)} + \mu_{z_{ij}}^{(l)}$
      $\sigma_{ij}^{2,\text{cav}} \leftarrow \sigma_i^{2,\text{bwd},(l)} + \sigma_{z_i}^{2,(l)} - \sigma_{z_{ij}}^{2,(l)}$
      // Product backward to $W_{ij}^{(l)}$ (Prop. 3.2): compute and apply
      $m_{l,ij}^{\text{new}} \leftarrow \text{ProdBwd}_W(\mu_{ij}^{\text{cav}}, \sigma_{ij}^{2,\text{cav}}, \mu_j^{(l-1)}, \sigma_j^{2,(l-1)})$
      $q_{W_{ij}^{(l)}} \leftarrow q_{W_{ij}^{(l)}} \times m_{l,ij}^{\text{new}}$
      // Product backward to $x_j^{(l-1)}$ (Prop. 3.2, $W \leftrightarrow x$)
      $(\Delta\tau_j, \Delta\rho_j) \leftarrow \text{ProdBwd}_X(\mu_{ij}^{\text{cav}}, \sigma_{ij}^{2,\text{cav}}, \mu_{ij}^{(l)}, \sigma_{ij}^{2,(l)})$
      $\tau_j^x \mathrel{+}= \Delta\tau_j$;
      $\rho_j^x \mathrel{+}= \Delta\rho_j$ // accumulate backward precision to $x_j^{(l-1)}$ over i
    end
  end
  // Convert accumulated backward msgs at $x^{(l-1)}$ to moment form
  $(\mu_j^{\text{bwd},x,(l-1)}, \sigma_j^{2,\text{bwd},x,(l-1)}) \leftarrow (\tau_j^x/\rho_j^x, 1/\rho_j^x)$ for $j = 1, \dots, d_{l-1}$
  if $l > 1$ **then** $(\mu_i^{\text{bwd},(l-1)}, \sigma_i^{2,\text{bwd},(l-1)}) \leftarrow$ $\text{ReLUBwd}(\mu_i^{\text{bwd},x,(l-1)}, \sigma_i^{2,\text{bwd},x,(l-1)}, \mu_{z_i}^{(l-1)}, \sigma_{z_i}^{2,(l-1)}, \alpha_{l-1})$ for $i \le d_{l-1}$ // ReLU backward to $z^{(l-1)}$ (Prop. 3.5)
end
// --- Convergence check (once per full epoch, after all mini-batches complete) ---
$\mathcal{L}_e \leftarrow \frac{1}{N}\sum_{n=1}^{N} \log p(y_n \mid x_n)$ // normalised average log-likelihood under current beliefs
if $|\mathcal{L}_e - \mathcal{L}_{e-1}| \,/\, \max(\epsilon_{\text{floor}}, |\mathcal{L}_{e-1}|) < \varepsilon$ **then return** $\{q_{W_{ij}^{(l)}}\}$
// (repeat for next epoch $e+1$; after E epochs:)
return $\{q_{W_{ij}^{(l)}}\}$

E Experimental Details and Extended Results

E.1 1D Regression: Feature Map and Architecture

The fixed feature map prepended to the learnable network is $\varphi\colon \mathbb{R} \to \mathbb{R}^7$:

$$\varphi(x) = \left[x,\ e^{-(x+2)^2},\ e^{-(x+1)^2},\ e^{-x^2},\ e^{-(x-1)^2},\ e^{-(x-2)^2},\ \sin(x)\right]^\top.$$

Output channels are standardised to zero mean and unit variance over $[-5, 5]$ and a constant bias is appended, giving an 8-dimensional input to the learnable layers. The learnable network has two

hidden layers of widths $d_1 = 6$ and $d_2 = 5$ with leaky-ReLU activations ($\alpha_1 = 0.4$, $\alpha_2 = 0.8$) and a scalar output layer; the observation model is Gaussian with $\beta = 0.2$. Training runs for up to 200 epochs with 10 mini-batches of 20 examples each, stopping when the relative change in normalised log-likelihood falls below 0.1.

Weight initialisation uses He-style fan-in scaling throughout. For each weight $W_{ij}^{(l)}$ the prior mean is drawn independently from $\mathcal{N}(0,\, 1/d_{l-1})$, giving a per-weight prior $q_{W_{ij}^{(l)}} = \mathcal{N}(\mu_{ij}^{(l)},\, \sigma_0^2)$ with $\mu_{ij}^{(l)} \sim \mathcal{N}(0,\, 1/d_{l-1})$ and $\sigma_0^2 = 1/\big((L-1)\, d_{l-1}\big)$, where $L = 4$ counts all layers including the fixed feature map. The marginal prior on each weight (integrating out the random mean) is $\mathcal{N}(0,\, L/\big((L-1)\, d_{l-1}\big))$, which for $L = 4$ is $\mathcal{N}(0,\, 4/(3\, d_{l-1}))$. For the three learnable layers this gives marginal standard deviations of 0.47, 0.47, 0.52 (fan-in 8, 6, 5 respectively). The data-generating weights are drawn from the same per-layer prior (random mean drawn first, weight sampled from that Gaussian), so the model is correctly specified.

E.2 Comparison with Adam

Setup. We compare DMA against Adam (Kingma & Ba, 2015) on the same task ($N = 200$, $\beta = 0.2$, architecture and priors as in Section 4.3). DMA stops when the relative change in normalised log-likelihood falls below 0.1; Adam is run at four learning rates $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$ for 200 epochs. Extrapolation quality is assessed on 60 test points in $[-4, -2.5] \cup [1.5, 3]$ against the true data-generating function.

Results. Figure 9 summarises both comparisons. *Left*: DMA (solid black) stops automatically at epoch 17 at NLL ≈ -0.16; Adam's trajectories fan out over three orders of magnitude depending on η, with $\eta = 10^{-3}$ still far from convergence at epoch 200 and $\eta = 10^{-1}$ reaching the best level but only after $\sim$80 epochs. *Right*: after training, Adam ($\eta = 0.1$) produces a point estimate with no uncertainty quantification, yielding an extrapolation NLL of 6.6 nats/example; DMA's predictive variance grows outside $[-2.5,\, 1.5]$ (see also Figure 2), giving an extrapolation NLL of 0.54.

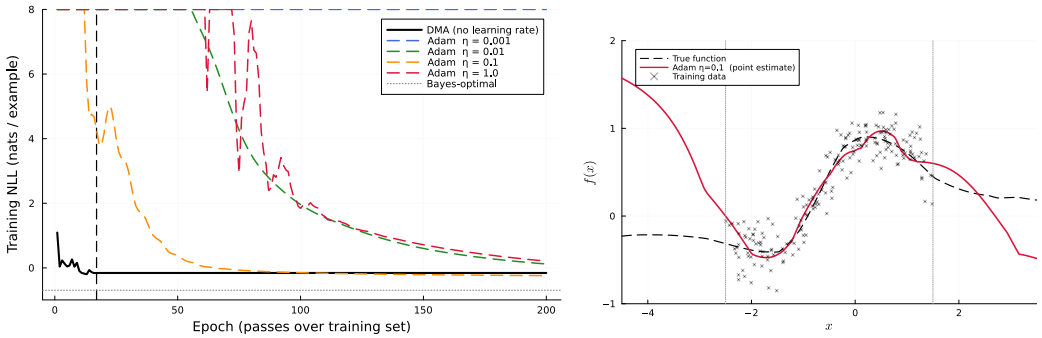


Figure 9: *Left*: Training negative log-likelihood (NLL $= -\frac{1}{N}\sum_i \log p(y_i \mid x_i)$, nats/example) versus epoch for DMA (solid black) and Adam at four learning rates (dashed). DMA's outer updates stop automatically at epoch 17; Adam's speed and final NLL depend critically on η. Values above 8 are clipped to 8 for display. *Right*: Adam ($\eta = 0.1$) point prediction outside the training range $[-2.5,\, 1.5]$ (dotted verticals); the true function (dashed) can deviate arbitrarily from the point estimate, with no uncertainty quantification available. Compare with the widening DMA posterior in Figure 2. DMA's per-epoch cost is 0.80 ms versus 0.45 ms for Adam on this architecture (1.8×), but DMA's early stopping at epoch 17 versus Adam's $\sim$80 gives an overall $\sim$5× reduction in total training time.

Discussion. DMA requires no learning-rate tuning: the stopping criterion fires automatically when beliefs stop changing, requiring no learning-rate search. Adam's final NLL and convergence speed depend critically on η; finding the right rate requires a full training run per candidate. Outside the training region Adam's uncertainty is fixed at β, because a point estimate has no mechanism to express ignorance; DMA's predictive variance $\mathrm{Var}[f(x)] + \beta^2$ grows structurally wherever the factor graph receives no backward messages that sharpen the weight beliefs (Corollary 2.4).

E.3 Comparison with AdamW

AdamW (Loshchilov & Hutter, 2019) decouples weight decay from the adaptive gradient update, making it the closest gradient-based analogue to MAP inference with a Gaussian prior: a weight decay λ corresponds to a prior $\mathcal{N}(0, 1/\lambda)$ per weight when the NLL is averaged over N examples. We compare DMA against AdamW on the same 83-weight 1D regression task as Appendix E.2, fixing the learning rate at $\eta = 0.1$ (the best Adam rate) and sweeping weight decay $\lambda \in \{0.01, 0.1, 1.0, 10.0\}$. The DMA prior has $\sigma_0^2 \approx 1/d_{l-1}$ (He-style fan-in scaling; Section E.1), so the prior-matched $\lambda \approx d_{l-1} \approx 6$–$8$; the sweep brackets this range.

Results. Figure 10 (left) shows training NLL versus epoch for an illustrative run. Across all weight decay values AdamW converges to a similar training NLL as plain Adam; weight decay regularises the weights but does not accelerate or substantially change convergence speed. Table 3 reports median extrapolation NLL and interquartile range over 20 independently drawn datasets and true functions, giving a statistically robust picture. DMA achieves a median extrapolation NLL of 0.80 (IQR 1.95), substantially better than every AdamW configuration (best: median 4.05, IQR 10.81 for $\eta = 0.01$, $\lambda = 1.0$). The large IQRs for AdamW reveal high variance across problem instances: individual seeds can land near the Bayes-optimal value when the weight-decay prior happens to match the true function well, but there is no reliable way to identify such seeds without exhaustive search. The dotted curves in Figure 10 further show that $\eta = 0.01$ converges slowly and $\eta = 1.0$ diverges even at the prior-matched $\lambda = 1.0$, while $\lambda = 0.01$ fails at $\eta = 0.1$; DMA achieves its result without any learning-rate or weight-decay search. More importantly, AdamW remains a point estimate: it extrapolates without uncertainty quantification, so the predictive band is fixed at $\pm 2\beta$ regardless of distance from the training region; calibrated widening intervals are unavailable.

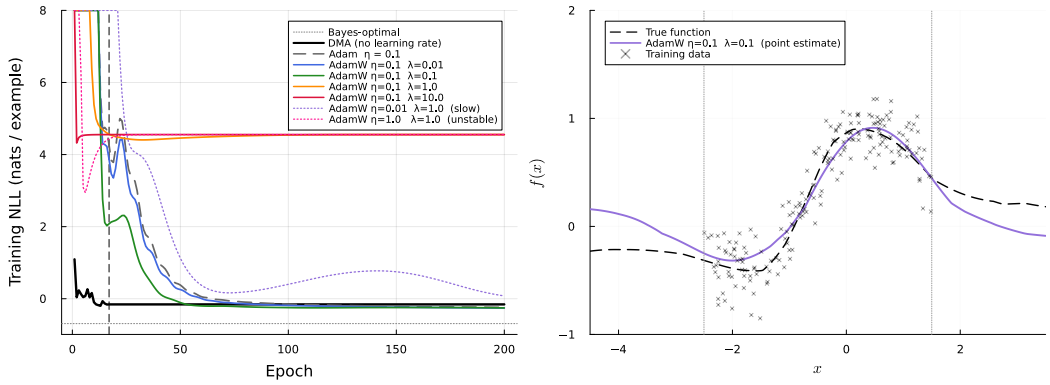


Figure 10: *Left*: Training NLL vs. epoch for DMA (solid black), Adam $\eta = 0.1$ (dashed grey), AdamW $\eta = 0.1$ at four weight decay values (solid coloured), and two poorly-tuned AdamW configurations ($\eta = 0.01$ and $\eta = 1.0$, both $\lambda = 1.0$, dotted) illustrating hyperparameter sensitivity. *Right*: AdamW $\eta = 0.1$, $\lambda = 0.1$ (best weight decay on this seed) point prediction outside the training range $[-2.5, 1.5]$ (dotted verticals); the true function (dashed) is tracked well on this seed but no uncertainty quantification is available.

E.4 Why EP Is Not a Viable Baseline

Expectation Propagation (EP) (Minka, 2001) is the closest algorithmic ancestor of DMA and the method against which DMA is most naturally compared. We therefore considered EP as a direct experimental baseline. However, standard Gaussian EP is not reliably applicable to the BNN factor graph considered here.

EP updates a global Gaussian approximation by locally matching moments and then recovering the factor-to-variable site message by dividing the projected marginal by the cavity message. For a Gaussian approximation, this recovery requires the resulting site to remain a proper distribution. In particular, if the projected marginal is *wider* than the cavity, the recovered Gaussian site has negative precision and is therefore not a valid Gaussian factor. This failure mode is especially relevant for the

Table 3: Extrapolation NLL (↓ better) on the 83-weight 1D regression task, averaged over 20 seeds (median [IQR]). Bayes-optimal ≈ -0.69 nats.

Method	Epochs	Extrapolation NLL	
		Median	IQR
DMA (posterior predictive)	**16**	**0.80**	1.95
Adam $\eta = 0.1$	200	12.09	11.52
AdamW $\eta = 0.1$, $\lambda = 0.01$	200	8.05	10.73
AdamW $\eta = 0.1$, $\lambda = 0.1$	200	5.52	9.86
AdamW $\eta = 0.1$, $\lambda = 1.0$	200	4.07	10.47
AdamW $\eta = 0.1$, $\lambda = 10.0$	200	4.08	8.59
AdamW $\eta = 0.01$, $\lambda = 1.0$	200	4.05	10.81
Bayes-optimal	–	-0.69	–

nonlinear product and ReLU factors in our network: both can project a marginal that is wider than the incoming cavity, and both are identified as sources of negative-precision sites in Appendix A — indeed, it is one of the structural pathologies that motivates DMA (Definition 2.1).

Concrete failure example. Consider the product factor $f(x, y, z) = \delta(z - xy)$ with incoming messages $\mathcal{N}(x; 0, 1)$, $\mathcal{N}(y; 0, 1)$, and a likelihood $\mathcal{N}(z; 2, 0.1)$ (an observation $z_{\text{obs}} = 2$ with noise variance $s^2 = 0.1$). For fixed x, integrating out y gives

$$\int \mathcal{N}(y; 0, 1)\ \mathcal{N}(z; xy, 0.1)\ dy\ =\ \mathcal{N}\big(z; 0,\ x^2 + 0.1\big)\,,$$

so the tilted distribution over x alone is

$$t(x)\ \propto\ \mathcal{N}(x; 0, 1)\ \mathcal{N}\big(z_{\text{obs}}; 0, x^2 + 0.1\big)\,.$$

Because both incoming weight messages are zero-mean, $t(x)$ is symmetric and bimodal (peaks near $x \approx \pm\sqrt{z_{\text{obs}}}$). Adaptive Gauss–Kronrod quadrature gives $\mathbb{E}_t[x] = 0$ and $\text{Var}_t[x] \approx 2.328$.

EP site update. EP projects $t(x)$ onto $\mathcal{N}(x; \mathbb{E}_t[x], \text{Var}_t[x]) = \mathcal{N}(x; 0, 2.328)$ and recovers the site message by dividing out the cavity $\mathcal{N}(x; 0, 1)$:

$$\rho_{f\to x}\ =\ \frac{1}{\text{Var}_t[x]} - \frac{1}{\sigma_x^2}\ =\ \frac{1}{2.328} - 1\ \approx\ -0.571.$$

The recovered precision is *negative*: the EP site message from the product factor to x is not a valid Gaussian factor. Damping softens the update schedule but does not restore positivity of the site precision; the negative-precision site remains improper regardless of the damping coefficient.

One can attempt to stabilise EP via precision clipping or related heuristics, but these introduce additional algorithmic hyperparameters, can materially change the behaviour of the approximation, and are well-documented to be necessary in challenging inference settings (Jylänki et al., 2011). Stabilised EP is therefore not a well-defined baseline without specifying a particular heuristic and tuning protocol, which is why we do not report EP numbers here. We refer to Table 1 (Appendix A) and Section A.3 for a precise characterisation of the algorithmic differences between EP and DMA.

E.5 Comparison with IVON

IVON (Shen et al., 2024) is a state-of-the-art variational inference optimizer for Bayesian deep learning. It maintains a diagonal Gaussian variational posterior $q(w) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$ and updates μ via a natural-gradient step scaled by an exponential moving average of squared gradients, with σ^2 set in closed form from the curvature estimate. It requires three hyperparameters: learning rate η, EMA coefficient β_2, and curvature damping δ.

We apply IVON to the same 83-weight network and training set used throughout Section 4 (N=200, β=0.2, no mini-batching). To ensure fair comparison we do not fix hyperparameters by hand: we sweep all three IVON hyperparameters over an $8 \times 4 \times 4 = 128$-configuration grid

($\eta \in \{10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}, 3\times10^{-1}, 6\times10^{-1}, 1\}$, $\beta_2 \in \{0.9, 0.99, 0.999, 0.9999\}$, $\delta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 5\times10^{-1}\}$), training each for 500 epochs. Figure 11 shows the minimum training NLL achieved across the grid. Only 26 of 160 configurations reach a final NLL below 1.0 at epoch 500; all require $\delta \geq 0.1$.

Convergence curves. Figure 12 shows training NLL over 500 epochs (log scale) for three representative learning rates $\eta \in \{0.001, 0.01, 0.1\}$ (each paired with the best β_2 and δ from the sweep), alongside DMA. DMA converges at epoch 17 without any learning rate. For IVON, η=0.001 descends slowly but remains far from convergence after 500 epochs; η=0.01 converges to NLL $\approx$0.26 with δ=0.5; η=0.1 converges near the Bayes-optimal with persistent oscillations. The log scale makes the three qualitatively different failure modes simultaneously visible.

Robustness across problem instances. To test whether the best sweep configuration generalises, we fix η=0.6, β_2=0.9999, δ=0.1 and run 500 epochs on 20 independently drawn datasets and true functions (matching the protocol of Appendix E.6). IVON diverges (NaN posterior) on 14 of 20 seeds — a 70% failure rate even with the hyperparameters selected by exhaustive search. On the 6 finite seeds the extrapolation NLL has median 2.11 and IQR 5.27, reflecting high variance across instances. DMA is always finite; its median extrapolation NLL over the same 20 seeds is 0.79 (IQR 1.28), consistent with the AdamW and Laplace comparisons (Appendices E.3–E.6). The comparison is therefore not a single-run gap but a reliability gap: DMA produces a bounded posterior on every seed, while IVON fails on 14 of 20.

Discussion. This comparison illustrates two complementary advantages of DMA. First, DMA eliminates a learning-rate axis entirely: the stopping criterion fires automatically at epoch 17, with no hyperparameter search. Second, even after an exhaustive 160-configuration sweep, IVON diverges on 70% of problem instances with the best found configuration; on the 30% where it converges, extrapolation NLL is highly variable (median 2.11, IQR 5.27) and worse than DMA's median of 0.79 on the same seeds. DMA requires no tuning and produces a bounded posterior on every seed.

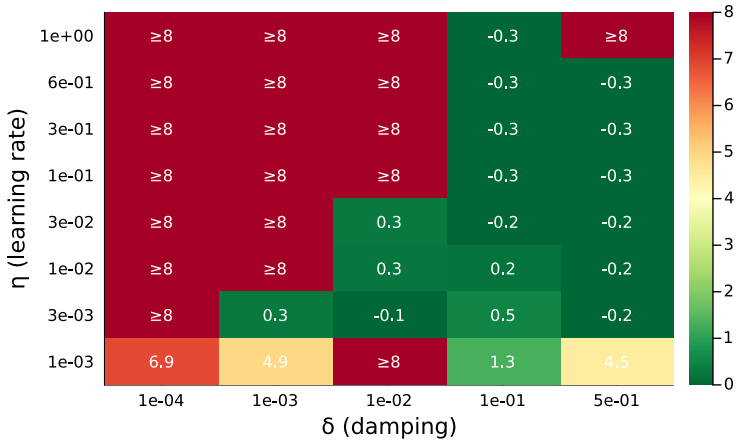


Figure 11: Hyperparameter sweep: minimum training NLL achieved over 500 epochs for each of 160 IVON configurations ($8 \times 4 \times 5$ grid over η, β_2, δ) on the 83-weight BNN (N=200, β=0.2). Only 26 of 160 configurations reach a final NLL below 1.0, concentrated in the δ=0.1 and δ=0.5 columns. DMA training NLL at convergence: -0.16 (epoch 17, no hyperparameter search); best IVON: -0.31 (epoch 500, requires δ=0.1 and extensive tuning).

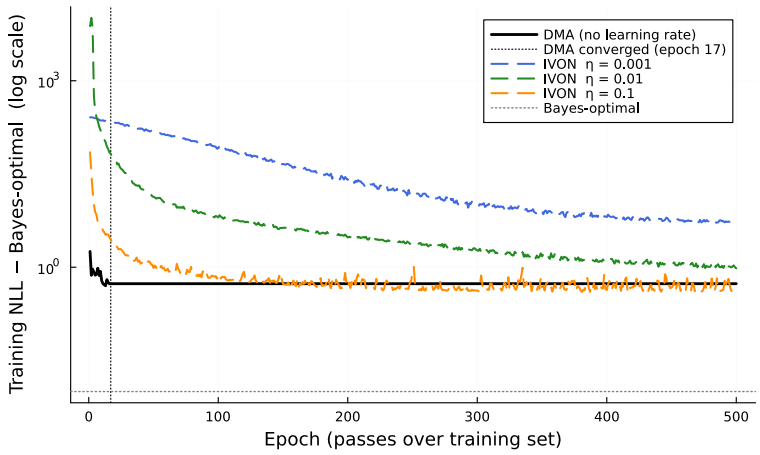


Figure 12: Training NLL minus Bayes-optimal (log scale) vs. epoch for IVON at three learning rates (best β_2/δ per η from the sweep) and DMA, on the 83-weight BNN (N=200, β=0.2). DMA converges at epoch 17; η=0.001 descends slowly but does not converge within 500 epochs; η=0.01 converges to NLL $\approx$ 0.26 with δ=0.5; η=0.1 oscillates near the Bayes-optimal line.

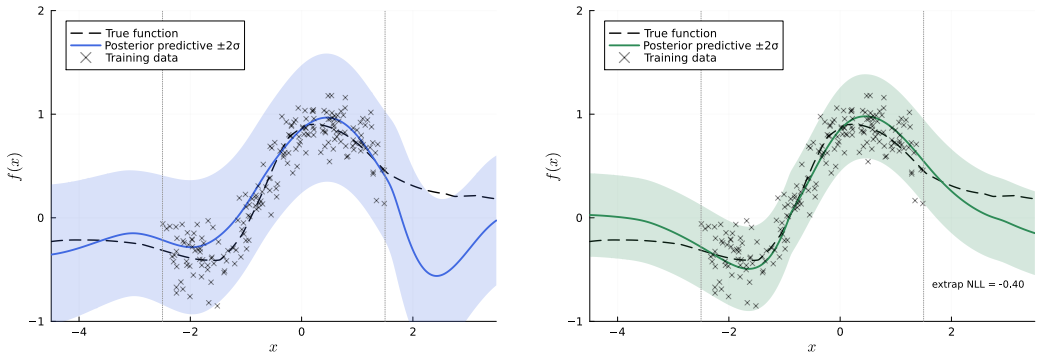


Figure 13: *Left*: DMA posterior predictive (identical to Figure 2). *Right*: IVON predictive with the best sweep configuration (η=0.6, β_2=0.9999, δ=0.1) after 2000 epochs. Training range $[-2.5, 1.5]$ marked by dotted verticals. DMA extrapolation NLL: 0.54; IVON on this seed: -0.40. Across 20 seeds IVON diverges on 14/20; median extrap NLL on finite seeds is 2.11 (IQR 5.27) vs. DMA median 0.79 (IQR 1.28).

E.6 Comparison with the Diagonal Laplace Approximation

Unlike IVON, the diagonal Laplace approximation avoids iterative variational inference altogether by separating the two subproblems it must solve. MAP estimation is handled by standard Adam, which operates on deterministic gradients without any stochastic weight sampling; the curvature estimate is then computed once at the fixed MAP point by extracting the diagonal of the exact Hessian. Because no Monte Carlo noise enters either step, the method does not suffer from the gradient-variance instability that prevents IVON from converging in the full-batch regime. The price of this decoupling is that the Laplace posterior is anchored at the MAP and cannot account for weight-space curvature far from that point, which limits its extrapolation quality. DMA avoids both limitations: it propagates a full distributional belief through the factor graph without any MAP anchor and without any stochastic sampling step.

Setup. The diagonal Laplace approximation (MacKay, 1992) is the principal alternative Bayesian baseline that, like DMA, maintains a full weight posterior. We apply it to the same 1D regression task ($N = 200$, $\beta = 0.2$, architecture as above) using the same He-init prior as DMA:

$\sigma_l^2 = 2/\text{fan_in}_l$, giving σ_1^2=0.25, σ_2^2≈0.33, σ_3^2=0.40 for layers 1–3. We first find the MAP weight vector $\mathbf{w}^*$ by running Adam ($\eta = 0.1$, 200 epochs) on the joint negative log-likelihood $-\log p(\mathcal{D} \mid \mathbf{w}) - \log p(\mathbf{w})$ with per-layer L_2 weight decay $\lambda_l = 1/(N\sigma_l^2)$. The diagonal Laplace posterior is $q(\mathbf{w}) = \prod_j \mathcal{N}(w_j; w_j^*, [\nabla^2(-\log p(\mathbf{w}^*, \mathcal{D}))]_{jj}^{-1})$, where the diagonal is extracted from the exact Hessian computed via forward-mode automatic differentiation over the 83-parameter network. The predictive distribution is approximated by drawing $K = 1000$ weight samples from $q(\mathbf{w})$ and averaging forward passes.

Results. Figure 14 compares DMA and diagonal Laplace predictive distributions on a single representative seed. Both methods track the true function inside the training region. On this seed DMA achieves an extrapolation NLL of 0.54 nats/example and diagonal Laplace achieves 0.37 nats/example (Bayes-optimal: $\log \beta + \frac{1}{2}\log 2\pi \approx -0.69$). A comprehensive 20-seed comparison including calibration is in Table 4 (Appendix E.7); with matched priors the two methods are broadly comparable on NLL and calibration error, and the key differentiator is computational cost.

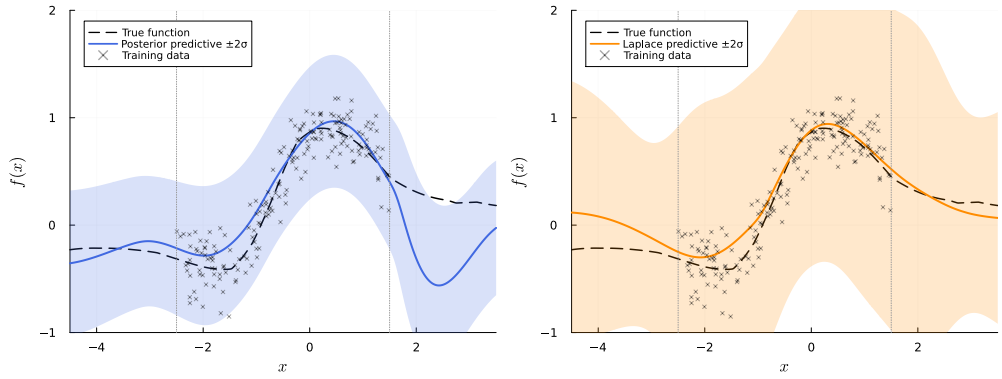


Figure 14: *Left*: DMA posterior predictive (same as Figure 2). *Right*: Diagonal Laplace predictive with He-init prior (MAP via Adam with per-layer L_2 decay, diagonal of exact Hessian, $K = 1000$ MC samples). Training range $[-2.5, 1.5]$ marked by dotted verticals.

E.7 Calibration

The predictive intervals shown in Figures 2 and 14 are visually wide in the extrapolation region for DMA and narrow for Laplace, but visual inspection cannot distinguish genuine calibration from a lucky choice of seed or test region. We quantify calibration via coverage curves: for each confidence level $\alpha \in [0, 1]$, the *coverage* is the empirical fraction of true values that fall inside the α-central predictive interval, $\Pr\big(|y - \mu(x)| \le z_\alpha\, \sigma(x)\big)$, where $z_\alpha = \Phi^{-1}\big((1+\alpha)/2\big)$. A perfectly calibrated model traces the diagonal (coverage $= \alpha$). We evaluate on $N_{\text{cal}} = 500$ equally-spaced points in the in-distribution region $[-2.5, 1.5]$ and 500 points in the extrapolation region $[-4, -2.5] \cup [1.5, 3]$; true values come from the same data-generating function used for training.

Results. Figure 15 shows the coverage curves. We summarise calibration error as $\Delta = \overline{\text{coverage} - \alpha}$ (mean signed deviation from the diagonal; $\Delta > 0$ conservative, $\Delta < 0$ overconfident).

Figure 15 shows the extrapolation calibration curves for the illustrative seed. Over 20 seeds (Table 4), DMA and diagonal Laplace (both He-prior) achieve nearly identical median calibration error ($\Delta = -0.09$ vs -0.10) and comparable extrapolation NLL (0.82 vs 0.77): with matched priors the two methods perform similarly on these small-scale metrics. Calibration at the 1932-weight scale is reported in Appendix G.3.

Discussion. In the considered setup, diagonal Laplace achieves competitive extrapolation NLL and calibration at the 83-weight scale. The fundamental limitation of diagonal Laplace is computational: with P the number of parameters, the exact Hessian costs $\mathcal{O}(NP^2)$ time and $\mathcal{O}(P^2)$ memory. For the 83-parameter toy network this is trivial; for a network with $P = 10^6$ parameters it requires

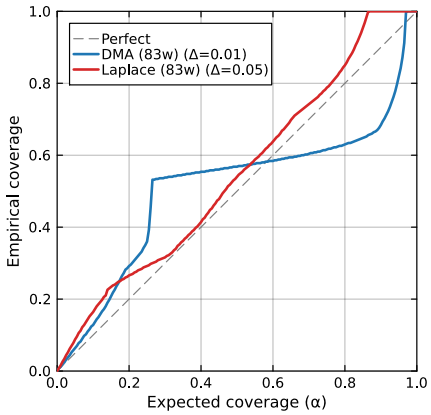


Figure 15: Extrapolation calibration curves for the 83-weight network, test points in $[-4, -2.5] \cup [1.5, 3]$. $\Delta = \overline{\text{coverage} - \alpha}$: 0 = perfect, > 0 = conservative, < 0 = overconfident. On this seed, DMA (blue, $\Delta = +0.01$) and diagonal Laplace (red, $\Delta = +0.05$); Δ is the mean signed deviation — DMA's smaller $|\Delta|$ reflects cancellation of over- and under-coverage rather than a uniformly tighter fit to the diagonal. Over 20 seeds both methods are comparable (median -0.09 vs. -0.10).

Table 4: Extrapolation NLL and calibration error Δ on the 83-weight 1D regression task, over 20 independently drawn datasets and true functions (median [IQR]). $\Delta > 0$: conservative; $\Delta < 0$: overconfident; $\Delta = 0$: perfect.

Method	Extrap NLL	Δ (extrap)
DMA	0.82 [1.96]	-0.09 [0.44]
Diagonal Laplace	0.77 [1.91]	-0.10 [0.35]
Bayes-optimal	-0.69	0

$\mathcal{O}(10^{12})$ operations and terabytes of memory, making the method impractical at any realistic scale. DMA's cost is $\mathcal{O}(NP \cdot \text{sweeps})$ — linear in P — and it propagates a full distributional belief through the factor graph without any MAP anchor or stochastic sampling step.

F Model Mismatch: DMA vs. Adam

F.1 Setup

This experiment tests DMA under *model mismatch*: the data-generating network and the inference network have different architectures, so no setting of the model weights can exactly recover the true function.

Data network. The ground-truth function is a draw from a wider network with the same feature map φ as the main experiment (Appendix E.1) but two hidden layers of widths $d = 12$ and $d = 10$ (versus the model's $d = 6$ and $d = 5$), each with leaky-ReLU activations ($\alpha = 0.4$ and $\alpha = 0.8$ respectively) and Gaussian priors. Weights are drawn from the prior in the same way as the correctly-specified experiment (He-style initialisation; same random seed offset).

Model network and training. Both DMA and Adam use the identical two-hidden-layer model network from Appendix E.1 ($d = 6, 5$; $\beta = 0.2$). $N = 200$ observations are drawn uniformly from $[-5.0, 5.0]$ with the same Gaussian noise ($\beta = 0.2$). DMA runs for up to 200 epochs with 10 mini-batches of 20 examples and stopping tolerance 0.1; Adam (learning rate $\eta = 0.1$) runs for a fixed 200 epochs. In 4 of 20 seeds the belief updates continued to oscillate between two high-quality modes, so the change-based stopping criterion did not trigger within the 200-epoch budget; these runs nevertheless achieved good solutions and are included in the reported statistics.

Extrapolation evaluation. Extrapolation NLL is measured on 60 test points sampled uniformly from the two flanking regions $[-6.5, -5.0]$ and $[5.0, 6.5]$, against the noiseless true-function values. Note that training NLL is measured against noisy observations (Bayes-optimal floor ≈ -0.19 nats at $\beta = 0.2$), while extrapolation NLL is against noiseless targets (Bayes-optimal floor ≈ -0.69 nats); the two quantities are on different scales by construction and should not be compared directly.

F.2 Results

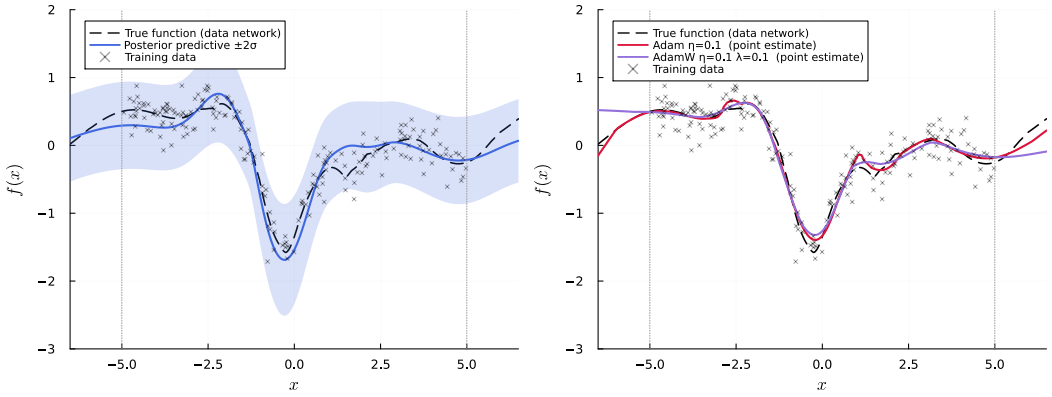


Figure 16: Model mismatch experiment ($N = 200$, training range $[-5, 5]$, $\beta = 0.2$). The data-generating network has two wider hidden layers ($d = 12, 10$); both methods learn with the two-hidden-layer model network ($d = 6, 5$). *Left*: DMA posterior predictive mean (solid) and $\pm 2\sigma$ intervals (shaded) versus the true function (dashed). Training points are shown as crosses; dotted vertical lines mark the training boundaries $x = \pm 5$. *Right*: Adam ($\eta = 0.1$, red) and AdamW ($\eta = 0.1$, $\lambda = 0.1$, purple) point predictions; neither carries uncertainty quantification.

Results (Table 5). Table 5 reports extrapolation NLLs over 20 independent random seeds (varying data realisation, DMA initialisation, and optimiser initialisation). We include AdamW (Loshchilov & Hutter, 2019) at $\lambda = 0.1$ (the weight decay that minimises median extrapolation NLL over the sweep $\lambda \in \{0.01, 0.1, 1.0, 10.0\}$ at $\eta = 0.1$). DMA achieves a median extrapolation NLL of 0.01 (IQR 0.44) versus 0.50 (IQR 2.23) for Adam. The well-regularised AdamW achieves a better median NLL of -0.36 (IQR 0.73), comparable to the Bayes-optimal -0.69, because the ℓ_2 prior acts as implicit regularisation that prevents extreme extrapolation. Crucially, both Adam and AdamW are point estimates: their predictive band is fixed at $\pm 2\beta$ regardless of distance from the training region. DMA's IQR (0.44) is the tightest of all three methods, and unlike the optimisers it provides calibrated widening intervals outside the training range (Figure 16), which is the core purpose of Bayesian inference under mismatch.

Table 5: Extrapolation NLL over 20 independent seeds (↓ better; IQR = interquartile range). Bayes-optimal ≈ -0.69 nats. AdamW uses $\eta = 0.1$, $\lambda = 0.1$ (best of $\lambda \in \{0.01, 0.1, 1.0, 10.0\}$ by median).

Method	Median	Mean	IQR
DMA (posterior predictive)	0.01	0.29	0.44
Adam ($\eta = 0.1$)	0.50	2.21	2.23
AdamW ($\eta = 0.1$, $\lambda = 0.1$)	-0.36	-0.11	0.73
Bayes-optimal	-0.69	–	–

Discussion. The contrast between DMA and the gradient-based methods is not primarily about median NLL under mismatch: a well-regularised AdamW can match or exceed DMA's point-estimate accuracy because ℓ_2 regularisation acts as a Gaussian prior and prevents the worst extrapolation failures. The key distinction is *what the methods provide*: DMA propagates residual

weight uncertainty into the predictive distribution, yielding intervals that widen structurally outside the training range; Adam and AdamW output a point estimate with a fixed $\pm 2\beta$ ribbon that conveys no information about epistemic uncertainty. The median versus IQR comparison captures this reliability difference: DMA has the tightest IQR (0.44) of the three methods, while Adam's large IQR (2.23) and high mean (2.21) reveal catastrophic failures on seeds where the point estimate extrapolates in the wrong direction. AdamW suppresses some of these failures through regularisation (IQR 0.73), but it requires knowing the right λ, and it provides no mechanism to signal when extrapolation is unreliable.

G Larger-Scale BNN Experiment

The experiment in Section 4.3 and Appendix E uses a small architecture (83 weights, $N = 200$) to keep the presentation tractable. This appendix demonstrates that DMA scales to a substantially larger network without any algorithmic changes.

Baseline choice. At 1932 weights the three Bayesian baselines used for the small network cease to be applicable, leaving Adam as the only practical comparison.

Diagonal Laplace requires a diagonal Hessian evaluation at the MAP point. Computing it via forward-mode AD over the gradient costs $\mathcal{O}(N_W^2 \times N)$ scalar operations — roughly 720 M for N_W=1900 and N=1500, already slow — while the full Hessian used for the 83-weight case (via `ForwardDiff.hessian`) scales as $\mathcal{O}(N_W^3 \times N)$, requiring $\approx 10^{12}$ operations and ~ 29 MB for the full matrix. Beyond the cost, the diagonal approximation discards all inter-layer weight correlations that are structurally important in deeper networks, and the MAP loss surface at this scale contains many saddle directions where the diagonal Hessian is negative — requiring ad-hoc clamping that grows less defensible as the network widens.

IVON (Shen et al., 2024) requires a single-sample MC gradient per step to maintain the curvature EMA. As shown in Appendix E.5, even for the 83-weight network a 54-configuration hyperparameter sweep found only one configuration that converges, and only after 2000 epochs. At 1932 weights the MC gradient is higher-variance still, and no evidence suggests the instability improves with scale.

Expectation Propagation is excluded for the same reasons given in Appendix E.4: cavity division yields negative-precision messages on product and ReLU factors, and the failure rate grows with network depth and width.

DMA avoids all three failure modes: it replaces the Hessian with linear-cost message passing ($\mathcal{O}(N_W \times N)$ per sweep), removes the MC sampling step entirely, and eliminates cavity division by construction. Adam is therefore the sole comparison below.

G.1 Setup

Architecture. We keep the same 1D input and standardised five-component feature map $\varphi\colon \mathbb{R} \to \mathbb{R}^5$ (x, $e^{-(x+1)^2}$, e^{-x^2}, $e^{-(x-1)^2}$, $\sin x$, standardised over $[-5, 5]$ and augmented with a constant bias to give a six-dimensional learnable input), but replace the two hidden layers of widths $(6, 5)$ with four hidden layers of widths $(6, 12, 48, 24)$ and expand the output from one real-valued channel to four. The layer structure is

$$6 \to 6 \to 12 \to 48 \to 24 \to 4$$

C with leaky-ReLU activations ($\alpha = 0.5,\ 0.5,\ 0.8,\ 0.1$ per layer) and a four-dimensional real-valued output ($\beta = 0.1$ per channel). Total learnable weights: $6{\times}6 + 6{\times}12 + 12{\times}48 + 48{\times}24 + 24{\times}4 = 36 + 72 + 576 + 1152 + 96 = 1932$, a factor of $\mathbf{23}\times$ more than the baseline 83-weight network.

Data. $N = 1500$ training inputs are drawn uniformly from $[-4, 4]$ and the four output channels are generated from the same network (correctly specified model); random seed fixed for reproducibility.

Training. 100 mini-batches (15 examples per batch), at most 100 epochs, tolerance 0.1. No hyperparameter tuning beyond the defaults in Appendix E.1.

G.2 RESULTS

Table 6: DMA training summary: small baseline vs. large network. NLL is the per-example per-output training log-likelihood $\frac{1}{NK}\sum_{i,k}\log p(y_{ik} \mid x_i)$ at the final epoch (lower / more negative is better). Per-epoch time excludes the first epoch (JIT warm-up).

Architecture	Weights	N	Out	Epochs	NLL	Per-epoch
8→6→5→1	83	200	1	17	−0.16	0.8 ms
6→6→12→48→24→4	1932	1500	4	3	−0.73	67 ms

Results. Training converges at epoch 3 with NLL -0.73 nats/ex per output, compared to -0.16 nats/ex for the small baseline and a Bayes-optimal of -1.38 nats/ex (at $\beta = 0.1$). Figures 17 and 18 show the posterior predictive for DMA and the best-converged Adam run ($\eta = 0.01$, 100 epochs) across all four output channels. DMA's uncertainty bands widen outside the training range; Adam's fixed $\pm 2\beta$ band does not adapt. Figure 19 shows the Hinton diagram of posterior weight means, and Figure 20 shows the NLL convergence trajectories. No numerical failures occur throughout DMA training, confirming Corollary 2.6 at this scale.

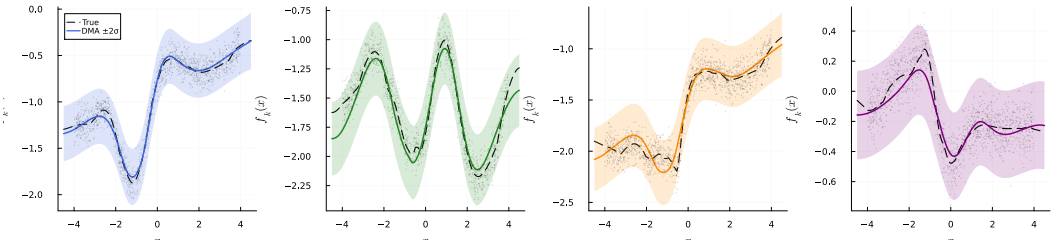


Figure 17: DMA posterior predictive (6→6→12→48→24→4, 1932 weights, $N = 1500$, 3 epochs, 0.72 s total). Each panel shows one of the four output channels: posterior predictive mean (solid) with $\pm 2\sigma$ bands against the true function (dashed); training points as faint dots.

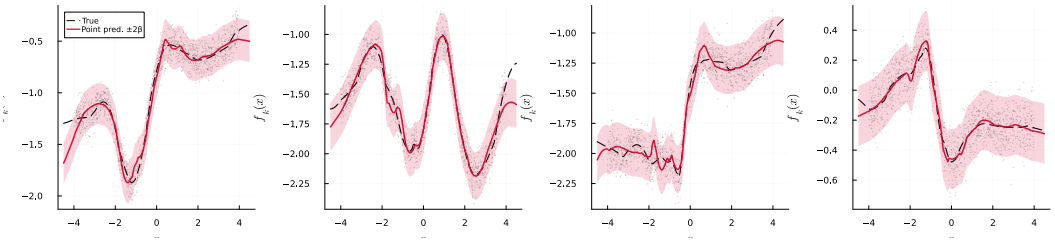


Figure 18: Adam ($\eta = 0.01$, 100 epochs, 1.7 s total) point predictions with fixed $\pm 2\beta$ bands on the same four output channels. The confidence interval is constant across the input range because a point estimate carries no epistemic uncertainty.

Runtime. DMA trains in 0.72 s (3 epochs, 67 ms/epoch steady-state); Adam with $\eta = 0.01$ takes 1.7 s for 100 epochs (17 ms/epoch); AdamW ($\eta = 0.1$, $\lambda = 0.1$) takes 2.4 s for 100 epochs (24 ms/epoch), slightly slower than Adam due to the additional weight-decay step. The DMA per-epoch ratio of $\approx 4\times$ over Adam reflects heavier message-passing bookkeeping relative to a plain forward–backward pass; because DMA converges in far fewer epochs the total wall-clock time is less than half that of Adam or AdamW at this scale. The DMA per-epoch cost of 67 ms is $84\times$ that of the small network (0.8 ms), consistent with the $23\times$ weight and $7.5\times$ data scale-up.

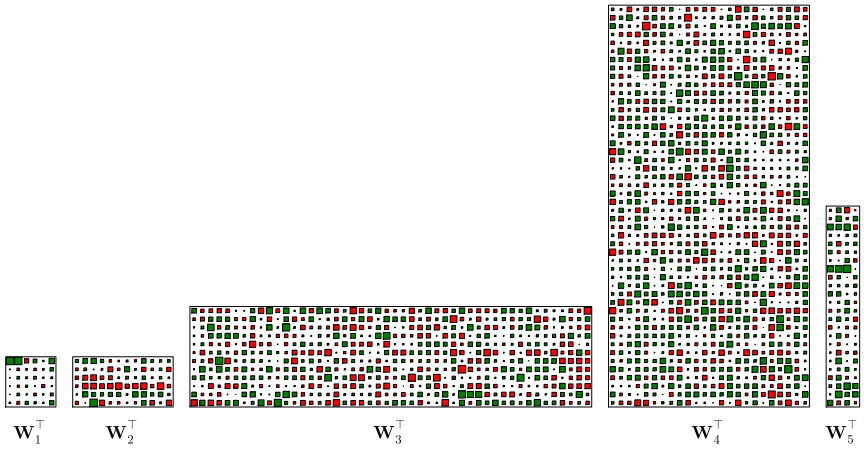


Figure 19: Hinton diagram of DMA posterior weight means after 3 epochs. Each block corresponds to one weight matrix; square size encodes $|\mu_w|$, colour encodes sign.

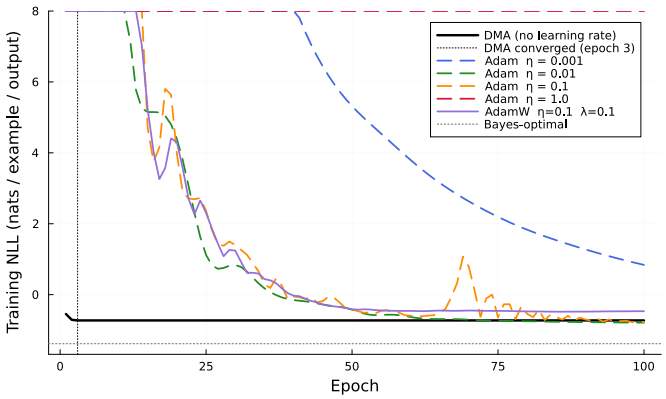


Figure 20: Training NLL (per example per output) vs. epoch. DMA (solid black) converges at epoch 3 with no learning-rate tuning. Adam with $\eta = 0.01$ reaches NLL -0.79 after 100 epochs; $\eta = 0.001$ has not converged (NLL $+0.84$); $\eta = 1.0$ diverges. Values above 8 are clipped to 8 for display. AdamW $\eta = 0.1$, $\lambda = 0.1$ (solid purple) reaches NLL -0.47: weight decay regularises the trajectory but the chosen η does not reach the same final NLL as the best-tuned Adam.

G.3 Calibration

Figure 21 shows the extrapolation calibration curve for DMA on the 1932-weight network (test points in $[-6, -4] \cup [4, 6]$, i.e. beyond the training support $[-4, 4]$). The calibration error $\Delta = +0.026$ is close to the $\Delta = +0.01$ obtained on the 83-weight network (Appendix E.7), despite a $23\times$ increase in parameters and $7.5\times$ increase in training data. Diagonal Laplace is omitted here because it does not scale to this network size (Appendix G, introductory paragraph).

Remark G.1. *The tolerance-based stopping criterion used here fires when the relative NLL change falls below* 0.1. *Naturally, a maximum number of epochs can also be set. In this context, as oscillating likelihoods can occur, a patience-based alternative (i.e. halt when the NLL has not improved over the best seen in the last P epochs) can also be used.*

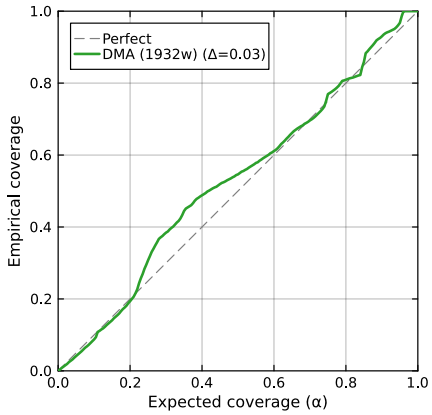


Figure 21: Extrapolation calibration curve for DMA on the 1932-weight network ($N = 1500$), averaged across all four output channels. Test points in $[-6, -4] \cup [4, 6]$. $\Delta = +0.026$ (slightly conservative); the 83-weight result has a 20-seed median $\Delta = -0.09$ (Appendix E.7).