Towards Knowledge-augmented Bayesian Deep Learning For Computer Vision

Supplementary Material

A. Theoretical Proofs

A.1. Proof of Theorem 3.1

Proof. Under Assumption 1, since the variational distribution space Ω is closed, there exists a limit point $\hat{p}^* \in \Omega$. We consider two cases and prove Eq. (13) under these cases separately.

- $\{\beta^{(k)}\}$ is bounded. In this case, based on step 6 in Algorithm 1, there exist k_0 such that $\beta_k = \beta_{k_0}$ for $\forall k \geq k_0$. This implies that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^{k_0+n}}[\phi^2(\mathbf{x},\boldsymbol{\theta})] \leq (\tau)^n \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^{k_0}}[\phi^2(\mathbf{x},\boldsymbol{\theta})]. \tag{16}$$

  When $n \to \infty$ and $0 < \tau < 1$, we have $\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}[\phi^2(\mathbf{x},\boldsymbol{\theta})] \to 0$. Hence Eq. (13) holds since $\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}[\phi^2(\mathbf{x},\boldsymbol{\theta})$ is always positive.
- $\beta^{(k)} \to \infty$ when k is sufficiently large. We try to prove it by contradiction. If there exist $q \in \Omega$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}[\phi^2(\mathbf{x},\boldsymbol{\theta})] > \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}[\phi^2(\mathbf{x},\boldsymbol{\theta})] \tag{17}$$

When $\beta^{(k)} \to \infty$ as $k \to \infty$, Eq. (17) implies

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \\ &> \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \end{aligned} \tag{18}$$

Hence, by the continuity of $\phi(\mathbf{x},\boldsymbol{\theta})$, there exists k_0 and $c > 0$ such that for $\forall k \geq k_0$,

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \\ &> \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] + c \end{aligned} \tag{19}$$

which equals to

$$\begin{aligned} &\frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \\ &> \frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] + \frac{\beta^{(k)}c}{2} \\ &\implies \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^*}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\ &\quad > \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + \frac{\beta^{(k)}c}{2} \end{aligned} \tag{20}$$

When the Dis function in Assumption 1 is the KL divergence, based on derivations in Appendix A.3, we have

$$\begin{aligned} &\text{Div}\left(\hat{p}^{(k)}||p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K}^{(k)})\right) = \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^{(k)}}[-\log p(\mathcal{D}|\boldsymbol{\theta})] \\ &+ \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^{(k)}}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + \text{KL}(\hat{p}^{(k)}||p(\boldsymbol{\theta})) + \text{const.} \end{aligned} \tag{21}$$

Then, we can rewrite Eq. (13) in Algorithm 1 by removing the constant term independent of q as follows:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^{(k)}}[-\log p(\mathcal{D}|\boldsymbol{\theta})] + \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^{(k)}}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\ &+ \text{KL}(\hat{p}^{(k)}||p(\boldsymbol{\theta})) \\ &\leq \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{D}|\boldsymbol{\theta})] + \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\ &+ \text{KL}(q||p(\boldsymbol{\theta})) + \epsilon_k \end{aligned} \tag{22}$$

For writing simplicity, we denote $f(q) = \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{D}|\boldsymbol{\theta})] + \text{KL}(q||p(\boldsymbol{\theta}))\ \forall q \in \Omega$, then when $k \to \infty$, Eq. (22) becomes:

$$\begin{aligned} &f(\hat{p}^*) + \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^*}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\ &\leq f(q) + \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + \epsilon_k \end{aligned} \tag{23}$$

However, starting from Eq. (20), we have

$$\begin{aligned} &f(\hat{p}^*) + \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^*}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\ &> f(q) + \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + f(\hat{p}^*) - f(q) + \frac{\beta^{(k)}c}{2} \\ &> f(q) + \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + \epsilon_k \end{aligned} \tag{24}$$

when $\beta^{(k)} \to \infty$ and ϵ_k is bounded. This is a contradiction to Eq. (23).

□

A.2. Proof of Theorem 3.2

Proof. Theorem 3.1 indicates the feasibility of $\hat{p}^*$ if the original optimization problem defined in Eq. (14) is feasible. That is $\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim p^*}[\phi^2(\mathbf{x},\boldsymbol{\theta})] = 0$. Let $q \in \Omega$ be a general feasible point such that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}[\phi^2(\mathbf{x},\boldsymbol{\theta})] = 0$, indicating $\phi(\mathbf{x},\boldsymbol{\theta}) = 0 \quad \forall x \in \mathcal{D}, \forall \boldsymbol{\theta} \sim q$. $\beta^{(k)} \to \infty$.

As $k \to \infty$, following the derivations for Eq.(23), the

Assumption 1 implies:

$$
\begin{aligned}
& f(\hat{p}^*) + \mathbb{E}_{\boldsymbol{\theta}\sim\hat{p}^*}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] \\
& \le f(q) + \mathbb{E}_{\boldsymbol{\theta}\sim q}[-\log p(\mathcal{K}^{(k)}|\boldsymbol{\theta})] + \epsilon_k \\
& \implies f(\hat{p}^*) + \frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \\
& \qquad \le f(q) + \frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}\left[\left(\phi(\mathbf{x},\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] + \epsilon_k
\end{aligned} \tag{25}
$$

Since both $\hat{p}^*$ and q are feasible, we have

$$
\begin{aligned}
& f(\hat{p}^*) + \frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim\hat{p}^*}\left[\left(\frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] \\
& \le f(q) + \frac{\beta^{(k)}}{2}\mathbb{E}_{\mathbf{x}\sim\mathcal{D},\boldsymbol{\theta}\sim q}\left[\left(\frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right] + \epsilon_k \\
& \implies f(\hat{p}^*) \le f(q) + \epsilon_k \implies f(\hat{p}^*) \le f(q)
\end{aligned} \tag{26}
$$

It is worth noting that minimizing $f(\cdot)$ is equivalent to minimizing the objective function in Eq.(14) as shown in Eq. (21). Since q is an arbitrary feasible point, $\hat{p}^*$ is the global minimizer. □

A.3. Proof of Theorem 3.3: Informative Knowledge Prior Tightens the PAC-Bayes Bound

We now show that the knowledge prior $p(\boldsymbol{\theta} \mid \mathcal{K})$ improves PAC-Bayes generalization guarantees compared to a baseline prior by reducing the KL term in the bound.

Let P be a data-independent prior over $\boldsymbol{\Theta}$ and Q a data-dependent posterior (e.g., an approximation to $p(\boldsymbol{\theta} \mid D, \mathcal{K})$). For bounded losses, a standard PAC-Bayes bound states that, with probability at least $1-\delta$ over the draw of D,

$$
\mathcal{R}(Q) \;\le\; \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{n}}{\delta}}{2(n-1)}}, \tag{27}
$$

where $\widehat{\mathcal{R}}(Q)$ is the empirical risk and $\mathcal{R}(Q)$ the true risk.

Let P_0 be a random baseline prior and $P_{\mathcal{K}}(\boldsymbol{\theta}) := p(\boldsymbol{\theta} \mid \mathcal{K})$ the knowledge prior from Stage 1. We assume that Q, P_0, and P_K have common support so that the KL divergences and expectations below are finite.

Lemma A.1 (KL decomposition for two priors)**.** *For any posterior Q,*

$$
\mathrm{KL}(Q\|P_{\mathcal{K}}) = \mathrm{KL}(Q\|P_0) - \mathbb{E}_{\boldsymbol{\theta}\sim Q}\Big[\log\frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\Big]. \tag{28}
$$

Proof. By definition,

$$
\begin{aligned}
\mathrm{KL}(Q\|P_{\mathcal{K}}) &= \int Q(\boldsymbol{\theta})\log\frac{Q(\boldsymbol{\theta})}{P_{\mathcal{K}}(\boldsymbol{\theta})}\,d\boldsymbol{\theta} \\
&= \int Q(\boldsymbol{\theta})\log\frac{Q(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\,d\boldsymbol{\theta} - \int Q(\boldsymbol{\theta})\log\frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\,d\boldsymbol{\theta}.
\end{aligned}
$$

The first integral is $\mathrm{KL}(Q\|P_0)$; the second is the stated expectation. □

Define the quantity

$$
\Delta(Q) := \mathbb{E}_{\boldsymbol{\theta}\sim Q}\Big[\log\frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\Big]. \tag{29}
$$

Then Lemma A.1 rewrites the PAC-Bayes KL term as

$$
\mathrm{KL}(Q\|P_{\mathcal{K}}) = \mathrm{KL}(Q\|P_0) - \Delta(Q).
$$

To show that the knowledge prior tightens the PAC-Bayes bound, we therefore need conditions under which $\Delta(Q) > 0$.

Step 1: Ideal case $Q = P_{\mathcal{K}}$. Consider first the "ideal" posterior $Q = P_{\mathcal{K}}$.

Lemma A.2 (Positivity of Δ at the knowledge prior)**.** *Assume $P_{\mathcal{K}}$ and P_0 have common support and $P_{\mathcal{K}} \neq P_0$. Then*

$$
\Delta(P_{\mathcal{K}}) := \mathbb{E}_{\boldsymbol{\theta}\sim P_{\mathcal{K}}}\Big[\log\tfrac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\Big] = \mathrm{KL}(P_{\mathcal{K}}\|P_0) > 0.
$$

Proof. Substituting $Q = P_K$ into (29) gives

$$
\Delta(P_{\mathcal{K}}) = \int P_{\mathcal{K}}(\boldsymbol{\theta})\log\frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\,d\boldsymbol{\theta} = \mathrm{KL}(P_{\mathcal{K}}\|P_0).
$$

By common support, the KL divergence is finite, and by $P_{\mathcal{K}} \neq P_0$ we have $\mathrm{KL}(P_{\mathcal{K}}\|P_0) > 0$. □

Thus, if the posterior equals the knowledge prior, the KL term in the PAC-Bayes bound is strictly smaller when we use $P_{\mathcal{K}}$ instead of P_0.

Step 2: Continuity of $\Delta(Q)$ around P_K. We now show that $\Delta(Q) > 0$ not only for $Q = P_{\mathcal{K}}$ but also for all posteriors Q that are *sufficiently close* to $P_{\mathcal{K}}$. For this we require a mild regularity condition on the log-density ratio.

Assumption 2 (Bounded log-density ratio)**.** There exists a constant $C_\pi < \infty$ such that

$$
\left|\log\frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\right| \;\le\; C_\pi, \qquad \forall\,\boldsymbol{\theta} \in \boldsymbol{\Theta}.
$$

This is satisfied, for example, when both $P_{\mathcal{K}}$ and P_0 are Gaussian with comparable covariance operators, or more generally when their densities are mutually bounded above and below by positive constants on $\boldsymbol{\Theta}$.

Under Assumption 2, the functional $Q \mapsto \Delta(Q)$ is Lipschitz-continuous with respect to total variation distance.

Lemma A.3 (Lipschitz continuity of $\Delta(Q)$)**.** *Let $\|\cdot\|_{\mathrm{TV}}$ denote total variation distance between probability measures. Under Assumption 2, for any two posteriors Q_1, Q_2,*

$$\big|\Delta(Q_1) - \Delta(Q_2)\big| \;\le\; 2C_\pi \, \|Q_1 - Q_2\|_{\mathrm{TV}}.$$

Proof. Let $f(\boldsymbol{\theta}) := \log \frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}$, so that $|f(\boldsymbol{\theta})| \le C_\pi$ by Assumption 2. Then

$$\Delta(Q_1) - \Delta(Q_2) = \int f(\boldsymbol{\theta}) \, \big(Q_1 - Q_2\big)(d\boldsymbol{\theta}).$$

By the variational characterization of total variation,

$$\begin{aligned}
&\big|\Delta(Q_1) - \Delta(Q_2)\big| \\
&\qquad \le \sup_{|g| \le C_\pi} \left| \int g(\boldsymbol{\theta}) \, \big(Q_1 - Q_2\big)(d\boldsymbol{\theta}) \right| \\
&\qquad \le 2C_\pi \, \|Q_1 - Q_2\|_{\mathrm{TV}},
\end{aligned}$$

where we used that $\sup_{|g| \le 1} \left| \int g \, d(Q_1 - Q_2) \right| = 2\|Q_1 - Q_2\|_{\mathrm{TV}}$. □

Combining Lemmas A.2 and A.3 yields the desired positivity of $\Delta(Q)$ in a full *neighborhood* of $P_{\mathcal{K}}$.

Proposition A.4 (Positivity of $\Delta(Q)$ near the knowledge prior)**.** *Assume common support, $P_{\mathcal{K}} \neq P_0$, and Assumption 2. Let*

$$\Delta_{\mathcal{K}} := \Delta(P_{\mathcal{K}}) = \mathrm{KL}(P_{\mathcal{K}} \| P_0) > 0.$$

Then for any posterior Q satisfying

$$\|Q - P_{\mathcal{K}}\|_{\mathrm{TV}} \;<\; \frac{\Delta_{\mathcal{K}}}{4C_\pi},$$

we have

$$\Delta(Q) \;\ge\; \frac{\Delta_{\mathcal{K}}}{2} \;>\; 0.$$

Proof. By Lemma A.3,

$$\big|\Delta(Q) - \Delta(P_{\mathcal{K}})\big| \le 2C_\pi \, \|Q - P_{\mathcal{K}}\|_{\mathrm{TV}}.$$

If $\|Q - P_{\mathcal{K}}\|_{\mathrm{TV}} < \Delta_{\mathcal{K}}/(4C_\pi)$, then

$$\big|\Delta(Q) - \Delta_{\mathcal{K}}\big| < \frac{\Delta_{\mathcal{K}}}{2},$$

which implies

$$\Delta(Q) > \Delta_{\mathcal{K}} - \frac{\Delta_{\mathcal{K}}}{2} = \frac{\Delta_{\mathcal{K}}}{2} > 0.$$

□

Proposition A.4 formalizes the idea that whenever the learned posterior Q stays sufficiently close to the knowledge prior $P_{\mathcal{K}}$, the expectation $\Delta(Q)$ is automatically positive. This is precisely the regime where knowledge plays a strong role in shaping the posterior.

Step 3: PAC-Bayes tightening. We can now state the PAC-Bayes tightening result as a corollary of Lemma A.1 and Proposition A.4.

Theorem (Restatement of Thm 3.3: Knowledge prior tightens PAC-Bayes bound)**.** *Suppose common support, $P_{\mathcal{K}} \neq P_0$, Assumption 2, and that the posterior Q lies in the neighborhood of $P_{\mathcal{K}}$ specified in Proposition A.4, so that $\Delta(Q) \ge \Delta_{\mathcal{K}}/2 > 0$. Then the PAC-Bayes bound (27) with prior $P = P_{\mathcal{K}}$ is strictly tighter than with $P = P_0$:*

$$\begin{aligned}
\mathcal{R}(Q) &\le \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P_0) - \Delta(Q) + \ln \frac{2\sqrt{n}}{\delta}}{2(n-1)}} \\
&< \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P_0) + \ln \frac{2\sqrt{n}}{\delta}}{2(n-1)}}.
\end{aligned}$$

Proof. Substitute Lemma A.1 into (27) with $P = P_{\mathcal{K}}$ to obtain

$$\mathcal{R}(Q) \le \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P_0) - \Delta(Q) + \ln \frac{2\sqrt{n}}{\delta}}{2(n-1)}}.$$

Since $\Delta(Q) \ge \Delta_{\mathcal{K}}/2 > 0$ and the square root is strictly increasing, this bound is strictly smaller than the corresponding bound with $P = P_0$ (where $\Delta(Q)$ is replaced by 0). □

Remark 1 (Feasibility of $\Delta(Q) > 0$ in practice)**.** The conditions for $\Delta(Q) > 0$ are naturally satisfied in many knowledge-aware settings:

- **Stage 1 (knowledge prior).** The prior $P_{\mathcal{K}}$ is typically obtained as a Laplace approximation around a knowledge-optimal parameter $\boldsymbol{\theta}_{\mathrm{prior}}$ (minimizer of a knowledge loss), making $P_{\mathcal{K}}$ sharply concentrated near a knowledge-feasible manifold $\mathcal{M}$.
- **Baseline prior.** The baseline prior P_0 is often chosen as a diffuse, isotropic Gaussian (or similar), which is much more spread out than $P_{\mathcal{K}}$. This implies $\mathrm{KL}(P_{\mathcal{K}}\|P_0) > 0$, and boundedness of the log-ratio is satisfied whenever the two priors have comparable tails (e.g., both Gaussian).
- **Stage 2 (hybrid posterior).** The hybrid posterior $p(\boldsymbol{\theta} \mid D, \mathcal{K})$ combines $P_{\mathcal{K}}$ with the data likelihood and an explicit knowledge penalty. When n is moderate and the knowledge term is strong, the resulting posterior Q remains close to $P_{\mathcal{K}}$ (in total variation or other weak metrics), placing it inside the neighborhood of Proposition A.4.

In this regime, $\Delta(Q)$ is automatically positive, and the PAC-Bayes bound with knowledge prior $P_{\mathcal{K}}$ is strictly tighter than with the uninformative baseline P_0. When the dataset is extremely large and the likelihood completely overwhelms the prior, Q may move far from $P_{\mathcal{K}}$ and $\Delta(Q)$ can approach zero; in that regime, the knowledge prior has little effect on the posterior, and we should not expect a tighter bound from knowledge alone.

A.4. Proof of Corollary 3.4: Knowledge Prior vs Small Normal Initialization

We now specialize the general result in Subsection A.3 to a common engineering baseline: a small isotropic Normal prior. This makes the conditions for $\Delta(Q) > 0$ concrete and interpretable.

Let

$$P_0(\boldsymbol{\theta}) = \mathcal{N}(0, \tau^2 I_d)$$

be the traditional BDL prior with small variance τ^2, and let $P_{\mathcal{K}}(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathcal{K})$ be the Stage 1 knowledge prior. In practice, $P_{\mathcal{K}}$ is taken as a Laplace approximation around a knowledge-optimal parameter $\boldsymbol{\theta}_{\text{prior}}$:

$$P_{\mathcal{K}}(\boldsymbol{\theta}) \approx \mathcal{N}\big(\boldsymbol{\theta}_{\text{prior}},\, \Sigma_{\text{prior}}\big),$$

where Σ_{prior} is the inverse Hessian of a knowledge loss at $\boldsymbol{\theta}_{\text{prior}}$.

To make the analysis clean, we adopt a standard bounded-parameter assumption.

Assumption 3 (Compact parameter set)**.** The parameter space $\boldsymbol{\theta} \subset \mathbb{R}^d$ is compact. Equivalently, we can view P_0 and $P_{\mathcal{K}}$ as Gaussians truncated to a large compact set $\boldsymbol{\theta}$, with densities renormalized on $\boldsymbol{\theta}$.

Under Assumption 3, all continuous functions of $\boldsymbol{\theta}$ are bounded on $\boldsymbol{\theta}$, including the log-density ratio $\log \frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}$.

Log-density ratio for Gaussian priors. For concreteness, write

$$P_{\mathcal{K}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, \Sigma_{\mathcal{K}}), \qquad P_0(\boldsymbol{\theta}) = \mathcal{N}(0, \tau^2 I_d),$$

with $\Sigma_{\mathcal{K}}$ positive definite and $\tau^2 > 0$ small. On $\boldsymbol{\Theta}$, their (unnormalized) densities are

$$p_{\mathcal{K}}(\boldsymbol{\theta}) \propto \exp\Big(-\tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}})^\top \Sigma_{\mathcal{K}}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}})\Big),$$
$$p_0(\boldsymbol{\theta}) \propto \exp\Big(-\tfrac{1}{2\tau^2}\|\boldsymbol{\theta}\|_2^2\Big).$$

Hence their log-density ratio is

$$\ell(\boldsymbol{\theta}) := \log \frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})} \tag{30}$$
$$= -\tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}})^\top \Sigma_{\mathcal{K}} {-} 1(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}}) + \tfrac{1}{2\tau^2}\|\boldsymbol{\theta}\|_2^2 + \text{const}, \tag{31}$$

where "const" is independent of $\boldsymbol{\theta}$ (it collects determinants and normalization factors).

We want to show that:

1. There is a region S around $\boldsymbol{\theta}_{\text{prior}}$ where $\ell(\boldsymbol{\theta})$ is strictly positive and bounded away from zero. 2. On the whole compact set $\boldsymbol{\Theta}$, $\ell(\boldsymbol{\theta})$ is bounded, so the bounded log-ratio assumption in Subsection A.3 holds. 3. The hybrid posterior Q from Stage 2 lies close to $P_{\mathcal{K}}$, so it assigns most of its mass to S, which makes $\Delta(Q) > 0$.

Lemma A.5 (Knowledge prior dominance in a neighborhood of $\boldsymbol{\theta}_{\text{prior}}$)**.** *Assume $\boldsymbol{\theta}_{\text{prior}} \neq 0$ and let $\lambda_{\max}$ denote the largest eigenvalue of $\Sigma_{\mathcal{K}}^{-1}$. There exists a radius $r > 0$ and a constant $\rho > 1$ such that, for the set*

$$S := \{\boldsymbol{\theta} \in \boldsymbol{\Theta} :\ \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}}\|_2 \le r\},$$

we have

$$\log \frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})} \ \ge\ \log \rho, \qquad \forall\, \boldsymbol{\theta} \in S.$$

Proof. Fix $r \in (0, \|\boldsymbol{\theta}_{\text{prior}}\|_2)$ and consider $\boldsymbol{\theta} \in S$. Using (31),

$$\ell(\boldsymbol{\theta}) = -\tfrac{1}{2}(\boldsymbol{\theta}{-}\boldsymbol{\theta}_{\text{prior}})^\top \Sigma_K^{-1}(\boldsymbol{\theta}{-}\boldsymbol{\theta}_{\text{prior}}) + \tfrac{1}{2\tau^2}\|\boldsymbol{\theta}\|_2^2 + \text{const}.$$

On S, we have

$$(\boldsymbol{\theta}{-}\boldsymbol{\theta}_{\text{prior}})^\top \Sigma_{\mathcal{K}}^{-1}(\boldsymbol{\theta}{-}\boldsymbol{\theta}_{\text{prior}}) \le \lambda_{\max}\|\boldsymbol{\theta}{-}\boldsymbol{\theta}_{\text{prior}}\|_2^2 \le \lambda_{\max} r^2.$$

Moreover,

$$\|\boldsymbol{\theta}\|_2 \ \ge\ \|\boldsymbol{\theta}_{\text{prior}}\|_2 - \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{prior}}\|_2 \ \ge\ \|\boldsymbol{\theta}_{\text{prior}}\|_2 - r.$$

Combining these,

$$\ell(\boldsymbol{\theta}) \ \ge\ -\tfrac{1}{2}\lambda_{\max} r^2 + \tfrac{1}{2\tau^2}(\|\boldsymbol{\theta}_{\text{prior}}\|_2 - r)^2 + \text{const}.$$

The right-hand side is a continuous function of r and τ^2. For fixed r with $0 < r < \|\boldsymbol{\theta}_{\text{prior}}\|_2$, as $\tau^2 \to 0$ the positive term $\frac{1}{2\tau^2}(\|\boldsymbol{\theta}_{\text{prior}}\|_2 - r)^2$ dominates, so the lower bound becomes arbitrarily large. Thus, for any desired $\log \rho > 0$, we can choose τ^2 sufficiently small (and r fixed) such that

$$\ell(\boldsymbol{\theta}) \ \ge\ \log \rho, \qquad \forall\, \boldsymbol{\theta} \in S.$$

This proves the claim. □

Lemma A.6 (Bounded log-density ratio on $\boldsymbol{\Theta}$)**.** *Under Assumption 3, the log-density ratio $\ell(\boldsymbol{\theta})$ in (31) is continuous on $\boldsymbol{\Theta}$ and hence attains its maximum and minimum. In particular, there exists $C_\pi < \infty$ such that*

$$\left|\log \frac{P_{\mathcal{K}}(\boldsymbol{\theta})}{P_0(\boldsymbol{\theta})}\right| \ \le\ C_\pi, \qquad \forall\, \boldsymbol{\theta} \in \boldsymbol{\Theta}.$$

Proof. The map $\boldsymbol{\theta} \mapsto \ell(\boldsymbol{\theta})$ is a polynomial (quadratic) plus a constant, hence continuous. On a compact set $\boldsymbol{\Theta}$, a continuous function attains its maximum and minimum; therefore its absolute value is bounded by some $C_\pi < \infty$. □

Lemma A.6 shows that the bounded log-ratio Assumption 2 from Subsection A.3 holds automatically for truncated Gaussian priors on a compact parameter set.

Posterior alignment with S. By construction, the knowledge prior $P_{\mathcal{K}}$ is concentrated in a neighborhood of $\boldsymbol{\theta}_{\text{prior}}$. In particular, for any $\varepsilon' > 0$, we can choose $r > 0$ such that

$$P_{\mathcal{K}}(S) \;\geq\; 1 - \varepsilon'.$$

If the Stage 2 posterior Q is close to $P_{\mathcal{K}}$ in total variation distance (as argued in Proposition A.4), then

$$Q(S) \;\geq\; P_{\mathcal{K}}(S) - \|Q - P_{\mathcal{K}}\|_{\text{TV}} \;\geq\; 1 - \varepsilon' - \|Q - P_{\mathcal{K}}\|_{\text{TV}}.$$

Thus, for any target $\varepsilon > 0$, we can choose ε' and a neighborhood of P_K (in total variation) such that

$$Q(S) \;\geq\; 1 - \varepsilon.$$

This matches the "posterior alignment" condition used in the generic argument of Subsection A.3.

Putting everything together, we obtain the following corollary.

Corollary (PAC-Bayes tightening vs small Normal prior)**.** Assume:

- The parameter set $\boldsymbol{\Theta}$ is compact (Assumption 3).
- $P_0(\boldsymbol{\theta}) = \mathcal{N}(0, \tau^2 I_d)$ with τ^2 sufficiently small.
- $P_{\mathcal{K}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, \Sigma_{\mathcal{K}})$ with $\boldsymbol{\theta}_{\text{prior}} \neq 0$.
- The learned posterior Q lies in a small total variation neighborhood of $P_{\mathcal{K}}$ (as in Proposition A.4).

Then there exists a set $S \subset \boldsymbol{\Theta}$, a constant $\rho > 1$, and a constant $C_\pi < \infty$ such that:

1. $P_{\mathcal{K}}/P_0 \geq \rho$ on S (Lemma A.5),
2. $|\log(P_{\mathcal{K}}/P_0)| \leq C_\pi$ on $\boldsymbol{\Theta}$ (Lemma A.6),
3. $Q(S) \geq 1 - \varepsilon$ for some $\varepsilon \in (0, 1)$ (posterior alignment).

Consequently, $\Delta(Q) = \mathbb{E}_Q[\log(P_{\mathcal{K}}/P_0)] > 0$, and the PAC-Bayes bound (27) with prior $P = P_{\mathcal{K}}$ is strictly tighter than with the small Normal prior P_0:

$$\mathcal{R}(Q) \leq \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\text{KL}(Q\|P_0) - \Delta(Q) + \ln\frac{2\sqrt{n}}{\delta}}{2(n-1)}}$$
$$< \widehat{\mathcal{R}}(Q) + \sqrt{\frac{\text{KL}(Q\|P_0) + \ln\frac{2\sqrt{n}}{\delta}}{2(n-1)}}.$$

Remark 2. Intuitively, a small Normal prior $P_0 = \mathcal{N}(0, \tau^2 I_d)$ is sharply peaked around the origin, while the knowledge prior $P_{\mathcal{K}}$ is sharply peaked around a knowledge-optimal $\boldsymbol{\theta}_{\text{prior}} \neq 0$. For sufficiently small τ^2, there is a region S around $\boldsymbol{\theta}_{\text{prior}}$ where $P_{\mathcal{K}}$ dominates P_0 by a large factor (Lemma A.5). The Stage 2 hybrid posterior Q inherits this concentration around $\boldsymbol{\theta}_{\text{prior}}$, so it places most of its mass where $P_{\mathcal{K}}/P_0$ is large. This makes $\Delta(Q)$ positive and ensures that, in this regime, using the knowledge prior $P_{\mathcal{K}}$ yields a strictly tighter PAC-Bayes bound than using the small Normal prior P_0.

B. Detailed Motivation for the Two-Stage Framework

In the main text, we propose a novel two-stage hybrid framework for knowledge-augmented Bayesian deep learning (BDL). This approach is designed to overcome the limitations of standard BDL methods, which typically fail to leverage valuable domain knowledge $\mathcal{K}$, and also to address the shortcomings of methods that rely only on an informed prior. Here, we provide a detailed discussion of why Stage 1 (learning an informative prior $p(\boldsymbol{\theta}|\mathcal{K})$) is **essential but not sufficient**, and why Stage 2 (employing an adaptive knowledge likelihood $p(\mathcal{K}|\boldsymbol{\theta}, \mathcal{D})$) is **necessary** for robust and reliable inference. Furthermore, we address the theoretical validity of incorporating knowledge in both stages, clarifying why this does not constitute "double counting".

B.1. Limitations of Standard BDL and Prior-Only Approaches

Standard Bayesian inference aims to find the posterior distribution of parameters $\boldsymbol{\theta}$ given data $\mathcal{D}$:

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{32}$$

In most BDL applications, $p(\boldsymbol{\theta})$ is chosen to be a simple, non-informative prior, such as an isotropic Gaussian $\mathcal{N}(0, \sigma^2 I)$. This approach is suboptimal for two main reasons:

1. It discards available domain expertise $\mathcal{K}$, which could constrain the parameter space and guide the model.
2. It often leads to poor sample efficiency, as the model must learn solutions that could have been provided by the prior.

A natural alternative is to first learn an informative prior $p(\boldsymbol{\theta}|\mathcal{K})$ and then use it for inference:

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{K})p(\boldsymbol{\theta}|\mathcal{K}) \tag{33}$$

We argue that this "prior-only" approach, while a significant improvement, is fundamentally insufficient for two key reasons: the **approximation problem** and the **"soft bias" (drift) problem**.

B.2. Stage 1: The Essential, but Insufficient, Knowledge Prior

Our Stage 1, which learns an informative prior $p(\boldsymbol{\theta}|\mathcal{K})$, is **essential**. As shown in our PAC-Bayes analysis (Theorem

3.3), using a learned $P_K = p(\boldsymbol{\theta}|\mathcal{K})$ as the prior provides a provably tighter generalization bound compared to a non-informative P_0. This pre-training effectively embeds domain knowledge into the model's initialization, providing a "crucial head-start" that, as shown in our experiments, dramatically improves sample efficiency in low-data regimes.

However, this stage is **insufficient** on its own due to two critical issues.

1. The Approximation Problem: The true, complete prior distribution $p(\boldsymbol{\theta}|\mathcal{K})$ that perfectly captures all knowledge is typically intractable. In practice, Stage 1 cannot find this full distribution. Instead, as described in Section 3.2, we find an imprecise point estimate (the MAP estimate) $\boldsymbol{\theta}_{prior}$ by optimizing a knowledge-based loss $\mathcal{L}_1$. We then *approximate* the prior as a Gaussian using the Laplace Approximation. This is, by definition, only a local, Gaussian approximation of the true, complex, and potentially multi-modal knowledge-constrained parameter space. Relying solely on this approximation as the only source of knowledge during inference is a significant compromise.

2. The "Soft Bias" and Gradient Drift Problem: Even if we had a perfect representation of $p(\boldsymbol{\theta}|\mathcal{K})$, a prior provides only a **soft inductive bias**. It defines the starting point for learning, but it does not guarantee that the final posterior will respect its constraints. This is due to the *gradient drift* phenomenon.

Consider the optimization objective (e.g., in Variational Inference):

$$\mathcal{L}_{VI} = \underbrace{\mathbb{E}_{q(\boldsymbol{\theta})}[-\log p(\mathcal{D}|\boldsymbol{\theta})]}_{\text{Data Likelihood}} + \underbrace{KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{K}))}_{\text{Prior Regularization}} \quad (34)$$

The data likelihood term is a sum over N training samples. As the dataset size N grows, the magnitude of the data gradient $\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{data}}\|$ scales proportionally. In contrast, the prior regularization gradient $\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{prior}}\|$ remains constant. Consequently, in regions where noisy data or spurious correlations conflict with the domain knowledge, the powerful data-driven gradients easily overwhelm the prior. This causes the posterior $q(\boldsymbol{\theta})$ to "drift" away from the knowledge manifold $\mathcal{M}$ into regions that satisfy the data but violate physical laws. This drift is unavoidable in standard Bayesian inference when N is large, rendering the prior effective only for initialization but useless for strict constraint enforcement.

B.3. Stage 2: The Necessary Adaptive Knowledge Likelihood

To counteract the approximation error and the gradient drift, Stage 2 is **necessary**. We re-introduce the knowledge $\mathcal{K}$ not as a static prior, but as a dynamic, **adaptive knowledge likelihood** $p(\mathcal{K}|\boldsymbol{\theta}, \mathcal{D})$. This leads to our full hybrid posterior formulation:

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{K}) \propto \underbrace{p(\mathcal{D}|\boldsymbol{\theta})}_{\text{Data Likelihood}} \cdot \underbrace{p(\mathcal{K}|\boldsymbol{\theta}, \mathcal{D})}_{\text{Knowledge Likelihood}} \cdot \underbrace{p(\boldsymbol{\theta}|\mathcal{K})}_{\text{Knowledge Prior}} \quad (35)$$

Addressing the "Double Counting" Concern. A natural question arises: *Does using $\mathcal{K}$ in both the prior and the likelihood constitute invalid Bayesian "double counting"?* We clarify that this is not redundant, but rather a consistent strategy to decouple the difficulty of knowledge satisfaction from data fitting. The two stages address mathematically distinct challenges:

- **Stage 1 (Prior → Global Topology via Projection):** Standard training methods typically attempt to optimize data likelihood and knowledge constraints *simultaneously* from a random initialization. This often leads to slow convergence as the optimization landscape is complex. In contrast, our Stage 1 isolates the knowledge constraint. It effectively performs a **projection** of the parameters into the "knowledge satisfaction area" (the feasible manifold $\mathcal{M}$) *before* data fitting begins. This ensures that the Bayesian inference in Stage 2 commences from a valid configuration, rather than wasting capacity learning constraints from scratch.
- **Stage 2 (Likelihood → Adaptive Balance):** Once initialized on the manifold, the data fitting begins. Here, the adaptive likelihood acts as the balancing mechanism. It dynamically adjusts the strength of the knowledge penalty (α, β) against the data likelihood. If the data gradient is strong and attempts to pull the parameters off the manifold (drift), the adaptive likelihood increases the penalty strength to compensate.

In summary, our framework offers a consistent approach to balancing knowledge and data: Stage 1 establishes the balance by projecting parameters onto the valid manifold, and Stage 2 maintains that balance by adaptively fighting gradient drift. The two stages are orthogonal and complementary: one provides the *initial projection* (prior), and the other provides the *dynamic correction* (likelihood).

B.4. Limitations of the Laplace Approximation in Stage 1

While Stage 1 provides a crucial "warm start" for inference, we acknowledge that the Laplace Approximation (LA) employs a Gaussian assumption $p(\boldsymbol{\theta}|\mathcal{K}) \approx \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, H^{-1})$, which inherently limits its capacity to model complex knowledge manifolds.

Unimodality vs. Multi-modality. Recent studies in Bayesian Deep Learning, such as [28], have demonstrated that the true posterior landscape of neural networks is often

highly non-convex and multi-modal. In the context of domain knowledge, the set of valid parameters $\mathcal{M}$ may also be multi-modal (e.g., in 3D pose estimation, multiple valid anatomical configurations may exist for a given occlusion). A single Gaussian approximation centered at the MAP estimate $\boldsymbol{\theta}_{\text{prior}}$ cannot capture these multiple modes and may under-represent the uncertainty in directions with low local curvature [11].

Hessian Approximation. Furthermore, computing the exact Hessian H for high-dimensional networks is computationally prohibitive. Following standard practice [11, 49], we rely on approximations (e.g., diagonal or K-FAC), which may introduce errors in estimating the covariance structure of the knowledge prior.

Justification. Despite these limitations, the Gaussian approximation in Stage 1 remains effective for our specific purpose: **initialization**. Unlike methods that rely *solely* on the prior for inference (where the Gaussian assumption would be a bottleneck), our framework uses the prior primarily to project the optimization into a single valid basin of attraction. Once initialized, the **Stage 2 Adaptive Likelihood** takes over. Crucially, the adaptive likelihood is not constrained to a Gaussian form; it penalizes violations pointwise during the optimization of $q(\boldsymbol{\theta})$. This allows the final posterior in Stage 2 to evolve beyond the initial Gaussian shape and explore the local non-convex structure of the manifold as driven by the data.

C. Claims and Details about Pre-training and Prior

C.1. Comparison with BANANA and Role of the Prior

While both our framework and BANANA [54] aim to incorporate domain knowledge into Bayesian Deep Learning (BDL) through a prior, the fundamental role of the learned prior and the inference mechanism differs significantly.

The Role of the Prior. In BANANA, the primary goal is to learn a sophisticated, expressive prior (using variational inference or flow matching) that fully captures the domain knowledge. In many of their settings, this learned prior is used directly for downstream tasks or serves as the heavy-lifting component of the inference. Consequently, BANANA requires a complex, computationally intensive training process to ensure the prior distribution $p(\boldsymbol{\theta} \mid \mathcal{K})$ is highly refined.

In contrast, our **Stage 1** views the informative prior $p(\boldsymbol{\theta} \mid \mathcal{K})$ primarily as a *knowledge-compliant initialization* (a "warm start"). We do not require this prior to be the final representation of the model. Instead, we aim to find a region in the parameter space that satisfies domain constraints to guide the *start* of the main Bayesian inference. The heavy lifting of strictly enforcing knowledge and fitting data occurs in **Stage 2**, where the *Adaptive Knowledge Likelihood* $p(\mathcal{K} \mid \boldsymbol{\theta}, \mathcal{D})$ dynamically corrects the trajectory.

Utilization of Unlabeled Data. Both methods utilize task inputs $\mathbf{x}$ (without labels $\mathbf{y}$) to encode knowledge. However, because we treat the prior as an initialization, our Stage 1 pre-training is exceptionally lightweight (e.g., 3–5 epochs), whereas BANANA typically requires full convergence of a variational objective.

C.2. Detailed Implementation of Stage 1 (Pre-training)

In Stage 1, our goal is to obtain a Maximum A Posteriori (MAP) estimate $\boldsymbol{\theta}_{\text{prior}}$ based solely on domain knowledge $\mathcal{K}$ and unlabeled data $\mathbf{X} = \{\mathbf{x}_i\}$, and subsequently construct a Gaussian approximation around it. This process establishes a "knowledge-compliant" initialization for the main Bayesian inference in Stage 2.

C.2.1. Pre-training Objective and Anti-Collapse Regularization

We define a primary knowledge loss $\mathcal{L}_{\text{knowledge}}(\mathbf{x}, \boldsymbol{\theta})$ that penalizes violations of domain constraints (e.g., geometric consistency, invariance, or joint limits). However, optimizing this loss without ground-truth labels $\mathbf{y}$ presents a fundamental challenge: **Representation Collapse**.

The Problem: Trivial Solutions. Most domain knowledge constraints considered in this work admit a trivial global minimum where the model maps all inputs to a single constant output, $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{c}$.

- **For Invariance Knowledge (Rotated MNIST, Grayscale CIFAR-10):** The constraint minimizes the difference between an input and its transformed version, $\|f_{\boldsymbol{\theta}}(\mathbf{x}) - f_{\boldsymbol{\theta}}(T(\mathbf{x}))\|^2$. A constant model $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{c}$ yields a loss of zero, perfectly satisfying the invariance, yet is useless for the task.
- **For Gradient Knowledge (Decoy MNIST):** The constraint enforces zero gradients on background pixels, $\|\nabla_{\mathbf{x}_{\text{bg}}} f_{\boldsymbol{\theta}}(\mathbf{x})\| = 0$. A constant model has zero gradients everywhere, again satisfying the constraint trivially.
- **For Biomechanical Knowledge (Hand Pose):** The constraint enforces outputs to lie within valid ranges. A model predicting the constant mean pose for all inputs satisfies this perfectly.

To prevent this collapse and ensure the learned prior is informative, we introduce **Metric Alignment** (for all tasks) and **Variance Regularization** (specifically for regression).

Metric Alignment. To prevent degenerate embeddings and ensure that the learned feature space preserves the geomet-

ric structure of the input data, we enforce an isometric consistency constraint. We assume that inputs distinct in the deep feature space should yield distinct outputs. Let $\mathbf{z}_i$ denote the feature embedding of input $\mathbf{x}_i$ (e.g., the penultimate layer output) and $\hat{\mathbf{y}}_i = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$ denote the predicted output (e.g., logits or joint angles). For randomly sampled pairs (i, j) within a mini-batch, we minimize:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(i,j)}\left[\left\|\mathbf{z}_i - \mathbf{z}_j\right\|_2 - \left\|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\right\|_2\right]^2. \tag{36}$$

This term acts as a topology-preserving regularizer. It effectively applies a "repulsive" force for distinct inputs, preventing them from collapsing to the same point $\mathbf{c}$, while allowing the knowledge constraints to shape the manifold of valid solutions.

Variance Regularization (Hand Pose). For the 3D hand reconstruction regression task, Metric Alignment alone may not be sufficient to encourage the model to span the *full* valid physical range. To maintain a non-degenerate distribution, we regularize the per-dimension variance of the predicted poses toward a target scalar determined by the valid joint-angle limits. Let σ_j^2 denote the empirical variance of the j-th predicted joint angle across the batch, and let $r_j = y_j^{\max} - y_j^{\min}$ be its valid physical range. The regularizer is:

$$\mathcal{L}_{\text{var}} = \frac{1}{D}\sum_{j=1}^{D}\left(\sigma_j^2 - (\alpha\, r_j)^2\right)^2, \tag{37}$$

where α is a target scaling fraction (set to 0.3). This forces the model to utilize a significant portion of the valid physical space, effectively preventing low-variance collapse during range-only training.

Total Stage 1 Objective. The final objective for finding the mode of our knowledge prior is:

$$\mathcal{L}_{\text{Stage1}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{knowledge}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \mathbb{I}_{\text{reg}}\cdot\lambda_{\text{var}}\mathcal{L}_{\text{var}} + \lambda_{\text{wd}}\|\boldsymbol{\theta}\|_2^2, \tag{38}$$

where $\mathbb{I}_{\text{reg}}$ is an indicator function for regression tasks. We train this for a minimal number of epochs (e.g., 3–5) to establish a valid initialization.

C.2.2. Label-Free Laplacian Approximation

Once we obtain the parameter $\boldsymbol{\theta}_{\text{prior}}$ that minimizes $\mathcal{L}_{\text{Stage1}}$, we construct the informative prior distribution $p(\boldsymbol{\theta}|\mathcal{K})$ using the Last-layer Laplace Approximation (LA).

Standard LA typically computes the Hessian of the negative log-likelihood of the labels ($-\log p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$). However, Stage 1 is label-free. Instead, we interpret the Stage 1 loss as the negative log-density of the unnormalized knowledge prior:

$$p^*(\boldsymbol{\theta} \mid \mathcal{K}) \propto \exp\left(-\mathcal{L}_{\text{Stage1}}(\boldsymbol{\theta})\right). \tag{39}$$

The "energy" of this system is defined entirely by the knowledge constraints and anti-collapse regularizers. We approximate the posterior around the mode $\boldsymbol{\theta}_{\text{prior}}$ as a Gaussian $\mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, \mathbf{H}^{-1})$. The precision matrix $\mathbf{H}$ is computed as the Hessian of the total pre-training objective:

$$\mathbf{H} = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_{\text{Stage1}}(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{prior}}}.$$

This Hessian captures the curvature of the **knowledge manifold**. Directions in parameter space with high curvature correspond to weights that are critical for maintaining physical consistency and metric alignment. By using $\mathbf{H}$ to define the covariance of the prior, we ensure that in Stage 2, the Bayesian inference is "stiff" (resistant to change) in directions that violate knowledge, but "flexible" in directions that are undetermined by knowledge.

C.3. Iterative posterior refinement in Stage 2

The iteration in Algorithm 1 is necessary because the knowledge term in Stage 2 is adaptive rather than fixed. In particular, the knowledge likelihood is parameterized by variables α and β, and therefore changes after each update:

$$p\Big(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D\Big) = \prod_{x\in D}\left[\sqrt{\frac{\beta^{(k)}}{2\pi}}\exp\left(-\frac{\beta^{(k)}}{2}\left(\phi(x,\boldsymbol{\theta}) + \frac{\alpha^{(k)}}{\beta^{(k)}}\right)^2\right)\right]. \tag{40}$$

Here, $\mathcal{K}^{(k+1)}$ does not denote a different knowledge source; it denotes the same knowledge $\mathcal{K}$ under the updated adaptive likelihood induced by $(\alpha^{(k)}, \beta^{(k)})$. Since changing (α, β) changes the effective likelihood term, the corresponding Stage 2 posterior also changes. Therefore, Stage 2 does not solve a single fixed posterior once, but instead solves a sequence of posteriors indexed by k.

More specifically, after updating the adaptive likelihood, the target posterior at iteration $k+1$ is

$$p(\boldsymbol{\theta} \mid D, \mathcal{K}^{(k+1)}) \propto p(D \mid \boldsymbol{\theta})\, p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)\, p(\boldsymbol{\theta} \mid \mathcal{K}), \tag{41}$$

where the prior term $p(\boldsymbol{\theta} \mid \mathcal{K})$ is the informative prior learned in Stage 1 and kept fixed throughout Stage 2. Hence, the role of the iteration is to progressively refine the posterior by tightening the adaptive knowledge likelihood, while preserving the Stage 1 knowledge prior.

Stage 1 learns a knowledge-induced prior

$$p(\boldsymbol{\theta} \mid \mathcal{K}) \approx \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, H^{-1}), \tag{42}$$

which provides a knowledge-consistent initialization and a useful inductive bias. However, a fixed prior alone cannot guarantee that the posterior will continue to satisfy the knowledge constraints after the data likelihood is introduced. During Bayesian inference, the task likelihood $p(D \mid \boldsymbol{\theta})$ may pull the solution toward regions that fit the data well but violate the knowledge. For this reason, the

knowledge must also appear as an adaptive likelihood term and be updated iteratively during Stage 2.

The process can therefore be viewed as follows. First, Stage 1 produces the prior $p(\boldsymbol{\theta} \mid \mathcal{K})$. Then, at iteration k, Algorithm 1 evaluates the current degree of knowledge violation under $\hat{p}^{(k)}$, updates $(\alpha^{(k)}, \beta^{(k)})$, constructs a new adaptive likelihood $p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)$, and defines the new target posterior in Eq. (41). Starting from the current posterior estimate $\hat{p}^{(k)}$, the posterior approximation method is then run again to obtain $\hat{p}^{(k+1)}$ for this updated target. This procedure is repeated until the expected violation is sufficiently small or the penalty parameter reaches its maximum.

C.3.1. How Algorithm 1 uses the Stage 1 output mathematically

Algorithm 1 takes the output of Stage 1 in two related ways. First, Stage 1 provides the mean and local curvature of the knowledge-induced prior through $\boldsymbol{\theta}_{\text{prior}}$ and H, yielding the Gaussian approximation in Eq. (42). Second, this prior is used to form the initial posterior estimate $\hat{p}^{(0)}(\boldsymbol{\theta} \mid D, \mathcal{K}^{(0)})$ and remains the fixed prior factor in every subsequent posterior target.

Concretely, the posterior estimator at iteration $k+1$ is obtained by

$$\hat{p}^{(k+1)}(\boldsymbol{\theta} \mid D, \mathcal{K}^{(k+1)}) = \arg\min_{q(\boldsymbol{\theta}) \in \Omega} \operatorname{Div}\Big(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid D, \mathcal{K}^{(k+1)})\Big), \tag{43}$$

while for sampling-based methods the update is implemented by warm-starting from the current posterior approximation and continuing inference under the new target posterior.

C.3.2. Reducing posterior inference to a tractable minimization

A central computational question is how the distributional optimization problem above can be solved in practice. The key observation is that the KL divergence, when chosen as the divergence measure Div, absorbs the intractable normalizing constant of the target posterior, reducing the problem to a standard minimization over q.

Concretely, let $Z^{(k+1)} = \int p(D \mid \boldsymbol{\theta})\, p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)\, p(\boldsymbol{\theta} \mid \mathcal{K})\, d\boldsymbol{\theta}$ denote the normalizing constant, which is generally intractable. The KL divergence expands as:

$$\begin{aligned}
&\operatorname{KL}\Big(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid D, \mathcal{K}^{(k+1)})\Big) \\
&= \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid D, \mathcal{K}^{(k+1)})}\, d\boldsymbol{\theta} && (44)\\
&= \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(D \mid \boldsymbol{\theta})\, p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)\, p(\boldsymbol{\theta} \mid \mathcal{K})}\, d\boldsymbol{\theta} && (45)\\
&\quad + \log Z^{(k+1)}. && (46)
\end{aligned}$$

Since $\log Z^{(k+1)}$ does not depend on q, minimizing over $q \in \Omega$ is equivalent to minimizing:

$$\begin{aligned}
\mathcal{L}(q) \;&= \mathbb{E}_q[-\log p(D \mid \boldsymbol{\theta})] \\
&\quad + \mathbb{E}_q\Big[-\log p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)\Big] \\
&\quad + \operatorname{KL}(q(\boldsymbol{\theta}) \,\|\, p(\boldsymbol{\theta} \mid \mathcal{K}))\,.
\end{aligned} \tag{47}$$

This objective is fully tractable: the first term is the expected negative data log-likelihood, the second enforces knowledge constraint satisfaction under the current $(\alpha^{(k)}, \beta^{(k)})$, and the third regularizes q toward the Stage 1 knowledge prior. Crucially, none of these terms require evaluating $Z^{(k+1)}$.

Substituting the Laplace approximation $p(\boldsymbol{\theta} \mid \mathcal{K}) \approx \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, H^{-1})$ into the KL regularizer, and expanding the adaptive knowledge likelihood from Eq. (40), the objective becomes:

$$\begin{aligned}
\mathcal{L}(q) \;&= \mathbb{E}_q[-\log p(D \mid \boldsymbol{\theta})] \\
&\quad + \mathbb{E}_q\left[\sum_{x \in D}\left(\alpha^{(k)}\phi(x, \boldsymbol{\theta}) + \frac{\beta^{(k)}}{2}\phi(x, \boldsymbol{\theta})^2\right)\right] \\
&\quad + \operatorname{KL}\big(q(\boldsymbol{\theta}) \,\|\, \mathcal{N}(\boldsymbol{\theta}_{\text{prior}}, H^{-1})\big)\,.
\end{aligned} \tag{48}$$

This expression shows explicitly how the Stage 1 output enters Stage 2: the Hessian H and mean $\boldsymbol{\theta}_{\text{prior}}$ from Stage 1 define a fixed quadratic regularizer that anchors the inference, while the iteration updates only the coefficients $(\alpha^{(k)}, \beta^{(k)})$ of the knowledge penalty across outer iterations. The specific method for minimizing $\mathcal{L}(q)$ over $q \in \Omega$, whether variational inference, MCMC, or ensemble-based approximations are discussed in Appendix D.

Therefore, Algorithm 1 should be understood as iteratively solving a sequence of posteriors with the same knowledge-induced prior and progressively tightened knowledge likelihood:

$$\hat{p}^{(0)} \;\to\; \hat{p}^{(1)} \;\to\; \cdots \;\to\; \hat{p}^{(k)} \;\to\; \hat{p}^{(k+1)}, \tag{49}$$

where each transition is driven by a new adaptive likelihood $p(\mathcal{K}^{(k+1)} \mid \boldsymbol{\theta}, D)$ and the prior $p(\boldsymbol{\theta} \mid \mathcal{K})$ is retained throughout the entire Stage 2 procedure.

D. Incorporating Knowledge into Posterior Estimation

With the knowledge representation, we will analyze how the knowledge likelihood $p(\mathcal{K}|\boldsymbol{\theta})$ affects the posterior estimation in Eq. (1). Since the exact posterior estimation is often intractable, the goal of this section is to show how $p(\mathcal{K}|\boldsymbol{\theta})$ can influence various approximate posterior estimation methods such as Ensemble, MCMC, and VI methods. In this paper, we mainly use the deep ensemble method [34]

as an approximate Bayesian inference backbone to incorporate domain knowledge, due to its superior performance in model averaging and uncertainty quantification compared to other Bayesian neural network approximations.

Ensemble Methods. Ensemble methods integrate multiple individual models to generate predictions, showcasing their effectiveness in accurately quantifying uncertainty and exhibiting robustness. Typically, these methods acquire Monte Carlo samples of neural network parameters during the inference process. The parameter samples are obtained by training the neural network multiple times. Incorporating domain knowledge, each individual ensemble component can be estimated as follows:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\min_{\boldsymbol{\theta}} -\log p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K}) \\ &= \arg\min_{\boldsymbol{\theta}} -\log p(\mathcal{D}|\boldsymbol{\theta}) - \log p(\mathcal{K}|\boldsymbol{\theta},\mathcal{D}) - \log p(\boldsymbol{\theta}|\mathcal{K})\end{aligned} \tag{50}$$

Ensemble methods differ mainly based on their design of the ensemble components and the training loss functions. Knowledge can be incorporated by adding a knowledge likelihood term to the loss functions. To use ensemble methods in Algorithm 1, we replace step 4 with $\boldsymbol{\theta}^{k+1} = \arg\min_{\boldsymbol{\theta}} -\log p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K}^{k+1})$, starting from the initialization at $\boldsymbol{\theta}^k$. Theorems 3.1 and 3.2 are general, considering a q distribution to approximate the true posterior. They can also be extended to ensemble methods by replacing the $q(\cdot)$ in the divergence measure with ensemble modes.

Other Bayesian Inference Methods. With knowledge $\mathcal{K}$, our proposed framework can also be applied to other Bayesian inference methods. For example, the traditional MCMC method can construct a Markov chain to directly sample from the unnormalized probability distribution, specifically $p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. Since only unnormalized probabilities are needed, the additional computation involves calculating the knowledge likelihood $p(\mathcal{K}|\boldsymbol{\theta})$ and combining it with $p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ for iterative sampling. While MCMC methods are often accurate, they are typically time-consuming, making them impractical for complex vision tasks. In contrast, VI aims to learn a variational distribution $q(\boldsymbol{\theta} \mid \boldsymbol{\psi})$, parameterized by $\boldsymbol{\psi}$, which is designed based on a specific distributional assumption to best approximate the true posterior distribution. It is more efficient than MCMC methods and typically achieves this approximation by minimizing the KL divergence between the two distributions. Following this approach, the knowledge-augmented VI for step 4 in Algorithm 1 is given by Eq. (51):

$$\begin{aligned}\boldsymbol{\psi}^* &= \arg\min_{\boldsymbol{\psi}} \mathrm{KL}\left(q(\boldsymbol{\theta} \mid \boldsymbol{\psi}) \,\|\, p(\boldsymbol{\theta} \mid \mathcal{D},\mathcal{K})\right) \\ &= \arg\min_{\boldsymbol{\psi}} \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}\left[-\log p(\mathcal{D} \mid \boldsymbol{\theta})\right] + \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}\left[-\log p(\mathcal{K} \mid \boldsymbol{\theta})\right] \\ &\qquad + \mathrm{KL}\left(q(\boldsymbol{\theta} \mid \boldsymbol{\psi}) \,\|\, p(\boldsymbol{\theta})\right).\end{aligned} \tag{51}$$

The derivation of Eq. (51) is provided below.

D.1. Derivation of Eq. (51)

To provide a detailed proof for the given equation, let's start with the first expression:

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\psi})||p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K})) \tag{52}$$

Here, $\mathrm{KL}(q||p)$ is the Kullback-Leibler (KL) divergence between the distributions q and p. The KL divergence is defined as:

$$\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\psi})||p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K})) = \int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\psi})}{p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K})}\, d\boldsymbol{\theta} \tag{53}$$

Using Bayes' theorem for the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K})$:

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K}) = \frac{p(\mathcal{D},\mathcal{K}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D},\mathcal{K})} \tag{54}$$

Thus, the KL divergence becomes:

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\psi})||p(\boldsymbol{\theta}|\mathcal{D},\mathcal{K})) = \int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\psi})}{p(\mathcal{D},\mathcal{K}|\boldsymbol{\theta})p(\boldsymbol{\theta})}\, d\boldsymbol{\theta} \\ + \log p(\mathcal{D},\mathcal{K})\end{aligned} \tag{55}$$

Since $\log p(\mathcal{D},\mathcal{K})$ is a constant with respect to $\boldsymbol{\psi}$, it can be dropped when taking the $\arg\min$ over $\boldsymbol{\psi}$:

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \left(\int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\psi})}{p(\mathcal{D},\mathcal{K}|\boldsymbol{\theta})p(\boldsymbol{\theta})}\, d\boldsymbol{\theta} \right) \tag{56}$$

Next, we use the fact that $p(\mathcal{D},\mathcal{K}|\boldsymbol{\theta}) = p(\mathcal{D}|\boldsymbol{\theta})p(\mathcal{K}|\boldsymbol{\theta})$:

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \left(\int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\psi})}{p(\mathcal{D}|\boldsymbol{\theta})p(\mathcal{K}|\boldsymbol{\theta})p(\boldsymbol{\theta})}\, d\boldsymbol{\theta} \right) \tag{57}$$

Separating the expectations, we obtain:

$$\begin{aligned}\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \Bigg(& \int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \log \frac{q(\boldsymbol{\theta}|\boldsymbol{\psi})}{p(\boldsymbol{\theta})}\, d\boldsymbol{\theta} \\ & + \int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \left(-\log p(\mathcal{D}|\boldsymbol{\theta})\right)\, d\boldsymbol{\theta} \\ & + \int q(\boldsymbol{\theta}|\boldsymbol{\psi}) \left(-\log p(\mathcal{K}|\boldsymbol{\theta})\right)\, d\boldsymbol{\theta} \Bigg)\end{aligned} \tag{58}$$

Thus, we have:

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \Big(\mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\psi})||p(\boldsymbol{\theta})) + \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}[-\log p(\mathcal{D}|\boldsymbol{\theta})] + \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}[-\log p(\mathcal{K}|\boldsymbol{\theta})] \Big) \tag{59}$$

This completes the proof of the given equation.

E. Extra Experiments: Gray CIFAR-10 with Grayscale-Invariance Knowledge

Experiment Settings and Domain Knowledge. To further evaluate the generality of our framework on natural image datasets, we conduct an additional experiment on the **grayscale CIFAR-10** dataset. Here, the domain knowledge encodes grayscale invariance—the semantic content of an image should remain unchanged when color information is removed. Formally, the knowledge constraint enforces that the model's prediction distribution should remain consistent before and after converting the input to gray scale:

$$K: \ \phi(x, \boldsymbol{\theta}) = \big\| p(y \mid x, \boldsymbol{\theta}) - p(y \mid g(x), \boldsymbol{\theta}) \big\| = 0, \tag{60}$$

where $g(x)$ converts a normalized RGB image x into its gray-scale counterpart by linearly weighting the RGB channels (0.299R + 0.587G + 0.114B) and replicating the result across channels. This constraint is incorporated through the adaptive likelihood term $p(K \mid \boldsymbol{\theta}, D)$ as defined in Eq. (7), with penalty parameters (λ, α, β) updated following Algorithm 1. We train an ensemble of five CNN models (SimpleCNN or ResNet-18 backbones) using the ALM-based adaptive likelihood with the following hyperparameters: learning rate 10^{-3}, weight decay 5×10^{-4}, batch size 256, and 100 training epochs with adaptive decay. All experiments use the standard CIFAR-10 training and test splits, with data augmentation identical to the color baseline.

Results. Table 9 reports the classification accuracy (ACC ↑), negative log-likelihood (NLL ↓), and knowledge constraint (KC ↓) metrics, while Table 10 presents the out-of-distribution (OOD) detection performance based on epistemic uncertainty, using SVHN and CIFAR-100 as OOD datasets. Consistent with earlier results on Decoy-MNIST and Rotated-MNIST, our **Likelihood-Only** variant already outperforms all Bayesian baselines across most metrics, showing that directly enforcing grayscale invariance effectively enhances model calibration and stability. The **Full** two-stage model further improves performance, achieving the highest accuracy (93.13%), lowest NLL (0.2209), and smallest knowledge constraint (0.1148). These results confirm that initializing from a knowledge-induced prior helps the adaptive likelihood converge faster and achieve stronger constraint satisfaction.

In OOD detection (Table 10), our models maintain competitive or superior robustness compared with BANANA and Gaussian-prior ensembles. In particular, the **Full** model achieves the best AUROC and AUPR on both CIFAR-10→SVHN (87.34/92.03) and CIFAR-10→CIFAR-100 (87.10/73.25), indicating that grayscale-invariance knowledge improves generalization to unseen color and texture domains. Overall, this experiment demonstrates that our knowledge-augmented Bayesian framework generalizes effectively to more complex natural-image settings, offering consistent gains in accuracy, calibration, and uncertainty reliability under distributional shifts.

F. Convergence Analysis for Alg. 1

In Table 11, we report the ACC, NLL, and Knowledge Satisfaction Constraint (KC) for each sub-optimization step in Algorithm 1 on the DecoyMNIST dataset. The results show that about three iterations are often sufficient to achieve significant improvements. Additional training may not be necessary, as it only leads to marginal gains.

Regarding time complexity, our method requires approximately double the training time compared to ensemble methods using Gaussian, Laplacian, or Logistic priors. After the first iteration of the sub-optimization in Algorithm 1, the time complexity is significantly reduced because we leverage previously trained parameters rather than starting from scratch. This approach requires fewer epochs for the model to converge in subsequent iterations, resulting in greater efficiency.

Compared to BANANA, which performs two variational inference tasks, our method's time complexity is similar when ensemble models are trained in parallel. Each variational inference task in BANANA is comparable to training a deterministic neural network (e.g., a single ensemble component). Thus, the overall time complexity of BANANA is approximately equivalent to training two deterministic neural networks. Similarly, our method doubles the training time of a single deterministic neural network when parallelization is used. It is also worth noting that while our method requires a longer training time, it achieves significantly better performance, justifying the additional computational cost.

Table 11. ACC, NLL, KC for each update step in Algorithm 1.

Method	Step 1	Step 2	Step 3	Step 4	Step 5
ACC	97.52	97.93	98.20	98.28	98.33
NLL	0.098	0.069	0.059	0.056	0.054
KC	0.094	0.052	0.021	0.012	0.010

G. Hyperparameter Sensitivity Analysis for Alg. 1

In Table 12, we show the hyperparameter analysis for $\alpha^{(0)}$ and $\beta^{(0)}$. It is shown that Algorithm 1 is not sensitive to its

Table 9. Performance on gray-scale CIFAR-10 for accuracy (ACC ↑), negative log-likelihood (NLL ↓), and knowledge constraint (KC ↓).

Method	ACC (↑)	NLL (↓)	KC (↓)
Gaussian Prior	92.31 ± 0.31	0.2357 ± 0.07	0.3618 ± 0.0451
Laplacian Prior	79.00 ± 0.54	0.6518 ± 0.08	0.4152 ± 0.0523
Logistic Prior	80.20 ± 0.43	0.6934 ± 0.09	0.4395 ± 0.0877
BANANA	92.88 ± 0.27	0.2314 ± 0.05	0.3049 ± 0.1923
Ours (Likelihood-Only)	92.67 ± 0.08	0.2273 ± 0.03	0.1226 ± 0.0090
Ours (Full)	**93.13 ± 0.06**	**0.2209 ± 0.00**	**0.1148 ± 0.0164**

Table 10. OOD detection results for gray-scale CIFAR-10 using epistemic uncertainty, with AUROC (↑) and AUPR (↑).

Method	CIFAR-10 → SVHN		CIFAR-10 → CIFAR-100	
	AUROC	AUPR	AUROC	AUPR
Gaussian Prior	86.29	89.89	85.44	69.66
Laplacian Prior	52.40	68.91	71.82	52.78
Logistic Prior	68.23	82.03	68.69	52.33
BANANA	87.16	91.02	86.49	71.79
Ours (Likelihood-Only)	83.49	87.28	86.68	71.94
Ours (Full)	**87.34**	**92.03**	**87.10**	**73.25**

hyperparameters within certain ranges.

Table 12. ACC and NLL for Decoy MNIST on our likelihood-only method with different $\alpha^{(0)}$ and $\beta^{(0)}$.

Method	ACC	NLL
$\alpha^{(0)} = 0.0001$ and $\beta^{(0)} = 0.0001$	98.33	0.054
$\alpha^{(0)} = 0.0002$ and $\beta^{(0)} = 0.0001$	98.29	0.050
$\alpha^{(0)} = 0.0003$ and $\beta^{(0)} = 0.0001$	98.45	0.048
$\alpha^{(0)} = 0.0001$ and $\beta^{(0)} = 0.0001$	98.32	0.067
$\alpha^{(0)} = 0.0001$ and $\beta^{(0)} = 0.0002$	98.26	0.062
$\alpha^{(0)} = 0.0001$ and $\beta^{(0)} = 0.0003$	98.35	0.056

H. Implementation Details

Here, we provide some implementation details of our experiments.

H.1. Decoy MNIST and Rotated MNIST

In these two tasks, we perform image classification using a simple 2-layer neural network with two fully connected layers, each with 50 nodes. We use a batch size of 64, a learning rate of 0.001, and train the model for 20 epochs on an NVIDIA GTX 2080 GPU. For stage 1, we only train 3 epochs; for stage 2, the primary loss function is cross-entropy loss, supplemented with a knowledge likelihood to train ensemble models. Each ensemble model has 5 components. All the experiments are repeated three times.

H.2. Implementation Details for 3D Hand Reconstruction

H.2.1. Network Architecture

Our monocular 3D hand reconstruction framework follows the standard MANO-based pipeline. A ResNet-50 backbone [24] extracts 2D image features from each RGB input of size 224×224. The feature map is passed to an iterative regression head [4] consisting of three fully connected layers with residual connections that predict the MANO parameters: pose $\mathbf{y}_{pose} \in \mathbb{R}^{15\times3}$ and shape $\mathbf{y}_{shape} \in \mathbb{R}^{10}$. The predicted parameters are used to generate a 3D hand mesh $\mathbf{P} \in \mathbb{R}^{21\times3}$ via the MANO layer [51].

H.2.2. Stage 1: Knowledge-Induced Prior Learning

Stage 1 pre-training enforces the biomechanics knowledge $\mathcal{K}$ from Eq. 15 by minimizing the violation while preventing naive solution with regularization terms.

The details of obtaining $p(\boldsymbol{\theta}|\mathcal{K})$ can be found in C.2.2.

H.2.3. Stage 2: Adaptive Likelihood Training

During the main Bayesian training stage, we combine the weak 2D supervision with the adaptive knowledge likelihood:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{knowledge}}. \tag{61}$$

Data term. For each image $\mathbf{X}$ with annotated 2D joints $\bar{\mathbf{p}}_{2D} \in \mathbb{R}^{21\times2}$, we assume Gaussian observation noise:

$$p(\bar{\mathbf{p}}_{2D}|\mathbf{X};\boldsymbol{\theta}) = \prod_i \mathcal{N}(\bar{\mathbf{p}}_{2D,i} \,|\, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2), \tag{62}$$

where the mean $\boldsymbol{\mu}_i$ is the weak-perspective projection of the predicted 3D joint $\mathbf{P}_i$ using camera parameters $\mathbf{C} = (s, t_x, t_y)$:

$$\boldsymbol{\mu}_i = s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{P}_i + (t_x, t_y)^\top. \tag{63}$$

The model directly regresses both $\mathbf{C}$ and the per-joint variance $\boldsymbol{\sigma}^2$. Minimizing the negative log-likelihood gives

$$\mathcal{L}_{\text{data}} = \sum_i \left(\log \boldsymbol{\sigma}_i + \frac{(\bar{\mathbf{p}}_{2D,i} - \boldsymbol{\mu}_i)^2}{2\boldsymbol{\sigma}_i^2} \right). \tag{64}$$

Knowledge term. The adaptive likelihood $\mathcal{L}_{\text{knowledge}}$ follows Algorithm 1, where the constraint variable $\phi(\mathbf{x}, \boldsymbol{\theta})$ from Eq. 15 is modeled by an augmented Gaussian:

$$p(\phi(\mathbf{x}, \boldsymbol{\theta})) = \sqrt{\tfrac{\beta}{2\pi}} \exp\left[-\tfrac{\beta}{2} \left(\phi(\mathbf{x}, \boldsymbol{\theta}) + \tfrac{\alpha}{\beta} \right)^2 \right], \tag{65}$$

and the corresponding loss is

$$\mathcal{L}_{\text{knowledge}} = -\sum_{\mathbf{x} \in D} \log p(\phi(\mathbf{x}, \boldsymbol{\theta})) \tag{66}$$

$$= \sum_{\mathbf{x} \in D} \left(\alpha\, \phi(\mathbf{x}, \boldsymbol{\theta}) + \tfrac{\beta}{2} \phi^2(\mathbf{x}, \boldsymbol{\theta}) \right). \tag{67}$$

The dual variables (α, β) are updated adaptively:

$$\alpha^{(k+1)} = \alpha^{(k)} + \beta^{(k)} \mathbb{E}[\phi^2],$$

$$\beta^{(k+1)} = \begin{cases} \beta^{(k)}, & \text{if } \mathbb{E}[\phi^2] < \tau \mathbb{E}_{prev}[\phi^2], \\ \gamma \beta^{(k)}, & \text{otherwise.} \end{cases}$$

H.2.4. Datasets and Evaluation

Training and evaluation use the FreiHAND dataset [7], containing 37 K training and 3.7 K test images. No 3D annotations are used—only 2D keypoints. To test robustness, we generate two occluded variants: FreiHAND-O and ARCTIC-O [14] by randomly erasing image regions following [69]. Evaluation metrics include: (1) 3D joint error E_J and vertex error E_V (mm) after Procrustes alignment, (2) knowledge-constraint violation $KC = \mathbb{E}_{x,\boldsymbol{\theta}}[\phi^2(x, \boldsymbol{\theta})]$, and (3) OOD detection AUROC/AUPR based on epistemic uncertainty.

H.2.5. Monocular 3D Hand Reconstruction

We implemented our framework using PyTorch. The hand images are scaled to 224×224 while preserving the aspect ratio. Training images are augmented with random scaling and horizontal flipping. The batch size is 64, and training proceeds until convergence. We use the Adam optimizer with a learning rate of 10^{-5}. All experiments are conducted on NVIDIA RTX A6000 GPUs.

H.3. Hyperparameters in Algorithm 1

The implementation details of step 4 in Algorithm 1 are illustrated in the previous section. In this section, we provide the hyperparameters used in Algorithm 1.

- $\alpha^{(0)}$: Initial value of α, chosen from {0.001, 0.0001, 0.00001}.
- $\beta^{(0)}$: Initial value of β, chosen from {0.001, 0.0001, 0.00001}.
- $\alpha^{\max}$: Maximum allowed value of α is 100.
- ϵ: Convergence threshold is 0.01 for Decoy MNIST, 0.001 for Rotated MNIST, and 0.3 for FreiHand.
- γ: Multiplicative factor for adaptive updating of β is 2.
- τ: Threshold for adaptive updating of β is 0.7.

I. Additional Ablations and Discussion

I.1. Stage-1-Only and Constrained Optimization Baselines

To further validate the necessity of each component in our two-stage framework, we evaluate two additional baselines on Decoy-MNIST.

Stage-1-Only Baseline. We evaluate a Prior-Only variant in which the Stage 1 pre-trained model is used directly for inference without any Stage 2 training. Since Stage 1 is entirely label-free (it optimizes only knowledge constraints and anti-collapse regularizers, as described in Appendix C.2), this variant has never seen any label supervision. As shown in Table 13, the Prior-Only model achieves only **12.89%** accuracy on Decoy-MNIST, which is near-random performance. This result confirms that while Stage 1 provides a knowledge-compliant initialization by projecting parameters onto the constraint manifold $\mathcal{M}$, it is Stage 2 that is essential for incorporating label supervision and preventing gradient drift away from $\mathcal{M}$.

Constrained Optimization Baseline. We compare against a standard deterministic constrained optimization baseline: a model trained from scratch with a fixed MSE constraint penalty (non-adaptive α, β). We also evaluate a variant initialized from our Stage 1 prior. As shown in Table 13, training from scratch with a fixed constraint achieves **85.7%** accuracy, and Stage-1 initialization improves this slightly to **86.4%**. Both significantly underperform our Likelihood-Only method (**98.33%**), which uses the same constraint but with our adaptive ALM-based likelihood. This validates that the adaptive update of (α, β) in Algorithm 1 is the critical mechanism driving effective constraint enforcement, not merely the presence of a constraint penalty.

I.2. Handling Non-Differentiable and Discrete Constraints

Our framework requires a differentiable constraint function $\phi(x, \boldsymbol{\theta})$, as it appears in the gradient of the adaptive likelihood. While this covers a broad class of constraints used in

Table 13. Accuracy (%) on Decoy-MNIST for additional baselines and ablations. Prior-Only uses Stage 1 model directly without label supervision. Fixed Constraint uses non-adaptive α, β. All other settings follow Table 1.

Method	ACC (%) ↑
Prior-Only (Stage 1, no labels)	12.89
MSE Constraint (from scratch)	85.70
MSE Constraint (Stage 1 init)	86.40
Ours (Likelihood-Only)	98.33
Ours (Full)	98.37

practice (geometric consistency, physical range limits, invariances), we discuss how discrete or logical constraints can be accommodated via standard continuous relaxations.

Logical constraints. For constraints expressed as logical rules, product t-norms provide differentiable approximations. For example, a conjunction $A \wedge B$ can be approximated as $A \cdot B$, and more complex rules follow analogously.

Conditional constraints with binary indicators. Consider the constrained problem:

$$\min_{\boldsymbol{\theta},\, x} \ \mathcal{L}(\boldsymbol{\theta}) \quad \text{subject to} \quad x\, g(\boldsymbol{\theta}) \le 0, \quad x \in \{0, 1\}, \tag{68}$$

which enforces $g(\boldsymbol{\theta}) \le 0$ when $x = 1$ and disables the constraint when $x = 0$. Relaxing x to $[0, 1]$, the augmented Lagrangian becomes:

$$\mathcal{L}(\boldsymbol{\theta}) + \alpha\, x\, g(\boldsymbol{\theta}) + \frac{\beta}{2}\big(x\, g(\boldsymbol{\theta})\big)^2, \tag{69}$$

which is smooth and directly compatible with Algorithm 1. This relaxation recovers the original conditional constraint in the limit of large α, β.

In general, any constraint that admits a differentiable relaxation, including soft logic, barrier functions, or penalty surrogates, can be incorporated into our framework without modification.