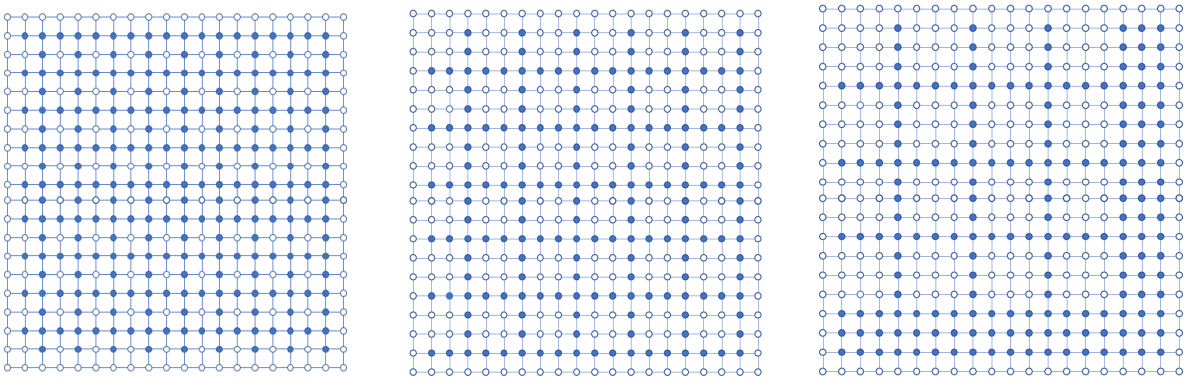
A Configurations for UAI Marginal MAP Experiments



(a) Combination 1 (b) Combination 2 (c) Combination 3

Figure 6: MAP/sum nodes combinations for UAI challenge datasets with shaded sum nodes and unshaded MAP nodes.

B Computational Details

B.1 Gradient of the Bethe Free Energy

Let η denote the parameters of the (approximate) marginals, then the negative Bethe free energy for a pairwise MRF can be expressed in terms of singleton and pairwise expectations, as follows:

$$\begin{aligned}
-\mathcal{F}(\eta) &= \sum_{i\in\mathcal{V}} \mathbb{E}_{b_i(x_i)}[\log \phi_i(X_i)] + \sum_{i\in\mathcal{V}} (1-N_i)\mathbf{H}[b_i] \\
&\quad + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{b_{ij}(x_i,x_j)}[\log \psi_{ij}(X_i,X_j)] + \sum_{(i,j)\in\mathcal{E}} \mathbf{H}[b_{ij}]) \\
&= \sum_{i\in\mathcal{V}} \mathbb{E}_{b_i(x_i)}[\log \phi_i(X_i) - (1-N_i)\log b_i(X_i)] + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{b_{ij}(x_i,x_j)}[\log \psi_{ij}(X_i,X_j) - \log b_{ij}(X_i,X_j)]
\end{aligned}$$

where N_i is the number of nodes adjacent to node i.

Using the fact that $\nabla_\theta \mathbb{E}_{p_\theta}[f(X)] = \mathbb{E}_{p_\theta}[f(X)\nabla_\theta \log p_\theta(X)] + \mathbb{E}_{p_\theta}[\nabla_\theta f(X)]$ for any distribution p_θ parameterized by θ and most real-valued functions f of interest, the (negative) gradient of the BFE can also be written as a sum of expectations:

$$\begin{aligned}
-\nabla_\eta \mathcal{F}(\eta) &= \sum_{i\in\mathcal{V}} \nabla_\eta \mathbb{E}_{b_i(x_i)}[\log \phi_i(X_i) - (1-N_i)\log b_i(X_i)] + \sum_{(i,j)\in\mathcal{E}} \nabla_\eta \mathbb{E}_{b_{ij}(x_i,x_j)}[\log \psi_{ij}(X_i,X_j) - \log b_{ij}(X_i,X_j)] \\
&= \sum_{i\in\mathcal{V}} \mathbb{E}_{b_i(x_i)}[\log \phi_i(X_i)\nabla_\eta \log b_i(X_i)] - (1-N_i)\mathbb{E}_{b_i(x_i)}[(\log b_i(X_i))\nabla_\eta \log b_i(X_i)] \\
&\quad + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{b_{ij}(x_i,x_j)}[\log \psi_{ij}(X_i,X_j)\nabla_\eta \log b_{ij}(X_i,X_j)] - \mathbb{E}_{b_{ij}(x_i,x_j)}[(\log b_{ij}(X_i,X_j))\nabla_\eta \log b_{ij}(X_i,X_j)] \\
&= \sum_{i\in\mathcal{V}} \mathbb{E}_{b_i(x_i)}[\{\log \phi_i(X_i) - (1-N_i)(\log b_i(X_i))\}\nabla_\eta \log b_i(X_i)] \\
&\quad + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{b_{ij}(x_i,x_j)}[\{\log \psi_{ij}(X_i,X_j) - \log b_{ij}(X_i,X_j)\}\nabla_\eta \log b_{ij}(X_i,X_j)]
\end{aligned}$$

The (negative) gradient of the BFE can be further simplified in the case of independent mixture beliefs for computational efficiency. In the most general case of a hybrid graphical model, let $\mathcal{V}_d, \mathcal{V}_c$ denote the set of discrete/continuous nodes,

respectively; for any node $i \in \mathcal{V}$, let $N_d(i)$, $N_c(i)$ denote the discrete/continuous neighbors of i, respectively. For any node i, let $b_i(x_i) = \sum_k w_k b_i^k(x_i)$, where $b_i^k(x_i)$ is the kth scalar component distribution. For a discrete node i with $x_i \in \mathcal{X}_i$, let b_i^k be a categorical distribution, such that $\sum_{x_i \in \mathcal{X}_i} b_i^k(x_i) = 1$. For a continuous node i, let η_{ik} be any scalar parameter of the kth component distribution $b_i^k(x_i)$. Then we have

$$\forall i \in \mathcal{V}_d, \forall x_i \in \mathcal{X}_i, -\frac{\partial \mathcal{F}}{\partial b_i^k(x_i)} = w_k\{\log \phi(x_i) - (1 - N_i)(\log b_i(x_i))\} + w_k \sum_{j \in N_d(i)} \mathbb{E}_{b_j^k(X_j)}[\log \psi(x_i, X_j) - \log b_{ij}(x_i, X_j)] + w_k \sum_{j \in N_c(i)} \mathbb{E}_{b_j^k(X_j)}[\log \psi(x_i, X_j) - \log b_{ij}(x_i, X_j)]$$

$$\begin{aligned}\forall i \in \mathcal{V}_c, -\frac{\partial \mathcal{F}}{\partial \eta_{ik}} &= w_k \mathbb{E}_{b_i^k(X_i)}[(\log \phi_i(X_i) - (1 - N_i)(\log b_i(X_i)))\, \frac{\partial}{\eta_{ik}} \log b_i^k(X_i)] \\ &+ w_k \sum_{j \in N_d(i)} \mathbb{E}_{b_i^k(X_i) b_j^k(X_j)}[(\log \psi(X_i, X_j) - \log b_{ij}(X_i, X_j))\, \frac{\partial}{\partial \eta_{ik}} \log b_i^k(X_i)] \\ &+ w_k \sum_{j \in N_c(i)} \mathbb{E}_{b_i^k(X_i) b_j^k(X_j)}[(\log \psi(X_i, X_j) - \log b_{ij}(X_i, X_j))\, \frac{\partial}{\partial \eta_{ik}} \log b_i^k(X_i)]\end{aligned}$$

For continuous nodes, the derivatives of (log) component beliefs $\frac{\partial}{\partial \eta_{ik}} \log b_i^k(x_i)$ are usually available in closed-form; e.g., if $b_i^k(x_i) = \mathcal{N}(x_i | \mu_{ik}, \sigma_{ik}^2)$, then $\frac{\partial}{\partial \mu_{ik}} \log b_i^k(x_i) = \frac{(x_i - \mu_{ik})^2}{\sigma_{ik}^2}$, and $\frac{\partial}{\partial \sigma_{ik}^2} \log b_i^k(x_i) = \frac{(x_i - \mu_{ik})^2 / (\sigma_{ik}^2) - 1}{2\sigma_{ik}^2}$.

B.2 Computing Expectations

As we saw earlier, computing the BFE and its gradient involves (approximately) computing expectations of various functions with respect to singleton and pairwise beliefs. By linearity of expectation, computing expectations with respect to mixture beliefs can be reduced to expectations with respect to the component distributions. Below we discuss the details of such computation in general, and show how it can be done in terms of basic matrix-vector operations.

B.2.1 Quadrature Approximation for Expectations with Respect to Continuous Beliefs

When X follows a normal distribution $\mathcal{N}(x | \mu, \sigma^2)$, the Gauss-Hermite quadrature (GHQ) rule can be used to give

$$\begin{aligned}\mathbb{E}[f(X)] &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-x^2) f(\sqrt{2}\sigma x + \mu) dx \\ &\approx \frac{1}{\sqrt{\pi}} \sum_{t=1}^{T} \lambda^{(t)} f(\sqrt{2}\sigma y^{(t)} + \mu)\end{aligned}$$

where the quadrature points $y^{(t)}$ and weights $\lambda^{(t)}$ are determined by the Gauss-Hermite method and are independent of the mean and variance.

Theorem 3 (Golub and Welsch (1969))**.** *For a positive integer T, mean $\mu \in \mathbb{R}$, and variance $\sigma^2 \in \mathbb{R}_{>0}$, GHQ constructs $\lambda^{(1)}, \ldots, \lambda^{(T)} \in \mathbb{R}$ and $y^{(1)}, \ldots, y^{(T)} \in \mathbb{R}$ such that there exists a $\xi \in \mathbb{R}$ with*

$$\mathbb{E}_{\mathcal{N}(\mu, \sigma^2)} f(x) = \sum_{t=1}^{T} \frac{\lambda^{(t)}}{\sqrt{\pi}} f\left(\sqrt{2\sigma^2} y^{(t)} + \mu\right) + \frac{n!\sqrt{\pi}}{2^n} \frac{f^{(2T)}(\xi)}{(2n)!}$$

As a consequence, using T quadrature points, the approximation is exact whenever f is a polynomial of degree at most $2T - 1$ in each variable separately.

In the bivariate case, assuming the joint distribution factorizes like $p(x_i, x_j) = p(x_i)p(x_j)$ so the dimensions are independent (which is assumed by our mixture beliefs, i.e., $b_{ij}^k(x_i, x_j) = b_i^k(x_i) b_j^k(x_j)$) , the 2-dimensional integral can be approximated by iterating the 1-dimensional quadrature rules:

$$\mathbb{E}[f(X_i, X_j)] \approx \frac{1}{\pi} \sum_{t=1}^{T} \lambda_i^{(t)} \sum_{s=1}^{T} \lambda_j^{(s)} f(\sqrt{2}\sigma_i y_i^{(t)} + \mu_i, \sqrt{2}\sigma_j y_j^{(s)} + \mu_j)$$

Similarly, the Gauss-Jacobi quadrature rule can be used to approximate expectations with respect to Beta distributions.

B.2.2 Expectations with Respect to Discrete/Mixed Beliefs

Let X be a discrete r.v. taking values in $\{1, 2, ..., S\}$, and let it follow a mixture of K categorical distributions,

$$p(x) = \sum_k w_k \pi_k(x) = \sum_k w_k \sum_{s=1}^{S} \pi_{k,s} \mathbf{1}(x = s)$$

where for each k, $\sum_{s=1}^{S} \pi_k(s) = \sum_{s=1}^{S} \pi_{k,s} = 1$. Then

$$\mathbb{E}_p[f(X)] = \sum_{s=1}^{S} \sum_{k=1}^{K} w_k \pi_{k,s} f(s) = \sum_{k=1}^{K} w_k \sum_{s=1}^{S} \pi_{k,s} f(s) = \boldsymbol{w}^\top \boldsymbol{\pi} f([1, \ldots, S]^\top)$$

Let X_i, X_j be discrete r.v.s taking values in $\{1, \ldots, S_i\}$ and $\{1, \ldots, S_j\}$, and follow a mixture of K independent categorical distributions,

$$p(x_i, x_j) = \sum_k w_k \pi_{ik}(x_i) \pi_{jk}(x_j)$$

Let $F = f\left([1, \ldots, S_i][1, \ldots, S_j]^\top\right) \in \mathbb{R}^{S_i \times S_j}$, then

$$\begin{aligned}
\mathbb{E}_p[f(X_i, X_j)] &= \sum_{s_i=1}^{S_i} \sum_{s_j=1}^{S_j} \sum_{k=1}^{K} w_k \pi_{ik}(s_i) \pi_{jk}(s_j) f(s_i, s_j) \\
&= \sum_{k=1}^{K} w_k \sum_{s_i=1}^{S_i} \pi_{ik}(s_i) \sum_{s_j=1}^{S_j} \pi_{jk}(s_j) f(s_i, s_j) \\
&= \sum_{k=1}^{K} w_k \boldsymbol{\pi}_{ik}^\top F \boldsymbol{\pi}_{jk} \\
&= \sum_{k=1}^{K} w_k \text{Tr}\left[\boldsymbol{\pi}_{jk} \boldsymbol{\pi}_{ik}^\top F\right] \\
&= \text{Tr}\left[\left(\sum_{k=1}^{K} w_k \boldsymbol{\pi}_{jk} \boldsymbol{\pi}_{ik}^\top\right) F\right] \\
&= \text{Tr}\left[\left(\boldsymbol{\pi}_j^\top \text{diag}(\boldsymbol{w}) \boldsymbol{\pi}_i\right) F\right] \\
&= \sum_{s_i, s_j} \left[\left(\boldsymbol{\pi}_i^\top \text{diag}(\boldsymbol{w}) \boldsymbol{\pi}_j\right) \odot F\right]_{s_i, s_j}
\end{aligned}$$

where $\boldsymbol{\pi}_i \in \mathbb{R}^{K \times S_i}$,$\boldsymbol{\pi}_j \in \mathbb{R}^{K \times S_j}$, and $\odot$ denotes element-wise product.

Finally, if X_i is discrete and X_j is continuous, so that

$$p(x_i, x_j) = \sum_{k=1}^{K} w_k \pi_{i,k}(x_i) p_j^k(x_j)$$

Then

$$\mathbb{E}_p[f(X_i, X_j)] = \sum_{k=1}^{K} w_k \sum_{s=1}^{S} \pi_{i,k}(s) \int p_j^k(x_j) f(s, x_j) dx_j = \sum_{k=1}^{K} w_k \sum_{s=1}^{S} \pi_{i,k}(s) \mathbb{E}_{p_j^k(X_j)}[f(s, X_j)]$$

where the inner expectations with respect to the scalar continuous distribution $p_j^k(X_j)$ can be approximated using quadrature methods as discussed above.

C Proof of Lemma 2

Claim. *Let A, B be two $n \times n$ real symmetric matrices, with B positive definite; let $\lambda_{min}(A)$ be the smallest eigenvalue of A. Then we have*

$$\text{Tr}[AB] \geq \lambda_{min}(A)\text{Tr}[B]$$

Proof. Denote the eigenvalues of A by $\lambda_1, ..., \lambda_n$, and denote the eigenvalues of B by $\gamma_1, ..., \gamma_n$. Let $A = U\Lambda U^T$, $B = V\Gamma V^T$ be the eigen-decompositions of A and B, such that U, V are orthogonal matrices, and Λ and Γ are diagonal matrices with $\Lambda_{ii} = \lambda_i$ and $\Gamma_{ii} = \gamma_i$ for $i = 1, ..., n$. Then

$$\begin{aligned}
\text{Tr}[AB] &= \text{Tr}[U\Lambda U^T V\Gamma V^T] \\
&= \text{Tr}[V^T U\Lambda U^T V\Gamma] \\
&= \text{Tr}[W^T \Lambda W\Gamma] && \text{let } W := U^T V \\
&= \text{Tr}[(\Lambda W)^T W\Gamma] \\
&= \sum_{i,j} \Lambda W \odot W\Gamma \\
&= \sum_i \langle \Lambda w_i, \gamma_i w_i \rangle && \text{let } w_i \text{ be the } i\text{th column of } W\text{; note that } \|w_i\| = 1 \text{ since } W \text{ is orthogonal} \\
&= \sum_i \gamma_i w_i^T \Lambda w_i \\
&\geq \sum_i \gamma_i \lambda_{min}(A) && \text{by the variational characterization of } \lambda_{min}(A), \text{ and } \gamma_i > 0 \\
&= \lambda_{min}(A)\text{Tr}[B]
\end{aligned}$$

□

D Computer Vision Experiments Setup

D.1 Model and Potentials Used in Optical Flow Experiment

Our model uses a robust penalty function (same as the Classic-C) called Charbonnier penalty, a differentiable variant of the L1 norm for both edge and node potentials. According to the additive property, we have $log\psi_{st} = \phi_{st}^{vert} + \phi_{st}^{hor}$, here we give the horizontal component as below:

$$\phi_{st}^{hor}(u_s, u_t) = -\lambda\sqrt{\epsilon^2 + (u_s - u_t)^2} \tag{7}$$

where λ is a regularization parameter determines the smoothness of optimized values spatially. Similarly the node potential enforces a data constancy of the given two images: matched part of object should have similar intensity. By using Charbonnier penalty again, we got node potential as follow:

$$\phi_s(u, v) = -\sqrt{\epsilon^2 + (I_1(i, j) - I_2(i + u, j + v))^2} \tag{8}$$

where I_1 and I_2 represent image intensity functions and we use bicubic interpolation to get a smooth function I_2 over continuous coordinate space. Our graphical model is a M×N×2 grid graph which can be visualized as Figure 7.

In the optimal flow experiment, we set the Charbonnier width $\epsilon = 0.001$ and regularization parameter $\lambda = 5$ for the pixel level comparison and $\lambda = 16$ for super pixel level comparison, to ensure consistency with competing approaches.

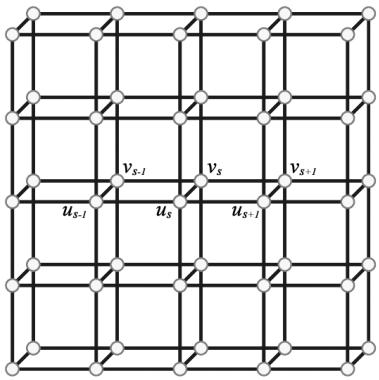


Figure 7: MRF model (a M×N×2 grid graph) for optical flow problem.

D.2 Model and Potentials Used in Stereo Depth Estimation Experiment

The graphical model for this problem is a simple M×N×1 grid, whose node potential is:

$$\phi(d_s) \propto \exp\left\{-\frac{1}{2\sigma^2}(I_L(i_s, j_s) - I_R(i_s + d_s, j_s))^2\right\} \tag{9}$$

where I_L represents the pixel value of left image and I_R represents the pixel value of right image. This term enforces the constancy of intensity of two images. Since we are treating d_s as continuous variables, here again bicubic interpolation was used to get a smooth I_R function in the continuous coordinate space.

For realistic scenes, the local evidence for a match can be weak, particularly in texture-less regions. We thus need another assumption to enforce the smoothness of neighboring pixels, which becomes the pairwise edge potential:

$$\psi(d_s, d_t) \propto \exp\left\{-\frac{1}{2\gamma^2}\min\left((d_s - d_t)^2, \delta_0^2\right)\right\} \tag{10}$$

D.3 More Details about Bicubic Interpolation

Bicubic interpolation is often better than bilinear or nearest-neighbor interpolation in terms of better smoothness. It considers 16 pixels (4×4) as sampling points in order to get the interpolated surface. MATLAB has a build-in function called `interp2` which implements bicubic interpolation when specified the 'cubic'parameter. For convenience and less overhead, we use MATLAB Coder to generate C code of this function and extract the core part to use it in our kernel function.