Problem 1: Fire Alarm Decision Network (20 points)

The following Bayes net is a modified version of the "Fire Alarm Decision Problem" from the Sample Problems of the Belief and Decision Networks tool on AIspace. All variables are binary, and the chance node CPTs as well as utility values are shown below.

Utility Table:

Fire	CheckSmoke	Call	Utility
True	True	True	-550
True	True	False	-850
True	False	True	-500
True	False	False	-800
False	True	True	-220
False	True	False	-20
False	False	True	-200
False	False	False	0

Chance Nodes:

Fire	Pr(Fire)
True	0.01
False	0.99

Fire	Alarm	Pr(Alarm | Fire)
True	True	0.95
True	False	0.05
False	True	0.01
False	False	0.99

(a) Compute and show the joint factor over all variables and the utility. Then eliminate any variable that is not a parent of a decision node, and show the resulting factor.

Resulting Joint Factor:

F	A	CS	C	utility
t	t	t	t	-5.225
t	t	t	f	-8.075
t	t	f	t	-4.750
t	t	f	f	-7.600
t	f	t	t	-0.275
t	f	t	f	-0.425
t	f	f	t	-0.250
t	f	f	f	-0.400
f	t	t	t	-2.178
f	t	t	f	-0.198
f	t	f	t	-1.980
f	t	f	f	0.000
f	f	t	t	-215.622
f	f	t	f	-19.602
f	f	f	t	-196.020
f	f	f	f	0.000

Table 1: Joint factor over all variables (F, A, CS, C) and the resulting utility.

(b) Sum out Fire:

Resulting Factor after Marginalizing Fire:

A	CS	C	utility
t	t	t	-7.403
t	t	f	-8.273
t	f	t	-6.730
t	f	f	-7.600
f	t	t	-215.897
f	t	f	-20.027
f	f	t	-196.270
f	f	f	-0.400

Table 2: Utility factor over (A, CS, C) after summing out Fire.

Maximum Expected Utility:

$$\boxed{\text{MEU}_{\text{no } F} = -7.13}$$

The optimal decisions are

$$CS^* = \text{false}, \qquad C^*(A) = \begin{cases} \text{true}, & A = \text{true} \\ \text{false}, & A = \text{false} \end{cases}$$

(c) Does Call depend on both parents?

Indeed, C^* depends *only* on A; its value is unchanged when CS toggles. Because CS^* is a constant, eliminating the decision nodes in any order yields the same policy.

(d) Observing *Fire* before making Call. With an additional informational arc $F \to C$ the best rule is

$$C(F) = \begin{cases} \text{true}, & F = \text{true} \\ \text{false}, & F = \text{false}, \end{cases} \qquad CS = \text{false}.$$

The expected utility improves to

$$\boxed{\text{MEU}_{\text{obs } F} = -5.00}.$$

(e) Value of Perfect Information on *Fire*.

$$\text{VPI}(F) = \text{MEU}_{\text{obs } F} - \text{MEU}_{\text{no } F} = (-5.00) - (-7.13) = \boxed{2.13}.$$

Hence direct knowledge of *Fire* is worth 2.13 utility units, because it prevents costly false alarms when A mis-fires.

Problem 2: Wandering Robot (20 points)

Consider the grid of states:

$$\begin{array}{ccc} A & B & C \\ D & E & \# \\ \# & F & G \end{array}$$

(a) Stationary Distribution: The robot's Markov chain is connected and aperiodic. Solving $\pi T = \pi$ yields, at equilibrium,

$$\pi_A = 2/14 \approx 0.1429, \quad \pi_B = 3/14 \approx 0.2143, \quad \pi_C = 1/14 \approx 0.0714, \pi_D = 2/14, \quad \pi_E = 3/14, \quad \pi_F = 2/14, \quad \pi$$

Equivalently,

$$\pi = (0.1429, 0.2143, 0.0714, 0.1429, 0.2143, 0.1429, 0.0714).$$

(b) State estimation with sensor readings (filtering).

The sensor returns one of three labels—"wall", "#", or "empty"—after looking in a uniformly random cardinal direction. The relevant likelihoods are

cell	A	B	C	D	E	F	G
$P(\text{wall} \mid X)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{2}$
$P(\# \mid X)$	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Starting from the stationary prior π and applying one transition, the predictive distribution is still π (because the chain is in equilibrium). Updating with the first observation $e_1 =$ "wall" gives

$$P(X_1 \mid e_1) = \left(\tfrac{4}{15},\ \tfrac{1}{5},\ \tfrac{2}{15},\ \tfrac{2}{15},\ 0,\ \tfrac{2}{15},\ \tfrac{2}{15}\right).$$

Next we perform a second transition and incorporate $e_2 = \#$:

$$P(X_2 \mid e_1, e_2) = \left(0,\ 0,\ \tfrac{1}{9},\ \tfrac{2}{9},\ \tfrac{1}{3},\ \tfrac{2}{9},\ \tfrac{1}{9}\right).$$

(c) Joint distribution and most-likely two-step paths.

For any $i, j \in \{A, \ldots, G\}$,

$$P(X_1 = i, X_2 = j \mid e_1, e_2) = P(X_1 = i \mid e_1)\ T_{ij}\ P(e_2 \mid X_2 = j),$$

which, after normalisation, is non-zero only for the pairs below:

$X_1 \to X_2$	$P(X_1, X_2 \mid e_1, e_2)$
$A \to D$	$\frac{2}{9}$
$G \to F$	$\frac{2}{9}$
$B \to C$	$\frac{1}{9}$
$B \to E$	$\frac{1}{9}$
$D \to E$	$\frac{1}{9}$
$F \to E$	$\frac{1}{9}$
$F \to G$	$\frac{1}{9}$

Hence the two most likely two-step trajectories are $\boxed{A \to D}$ and $\boxed{G \to F}$, each with posterior probability 2/9.

(d) Particle-filter localization:

We initialize one particle in each free cell $\{A, B, C, D, E, F, G\}$ and apply a single transition step according to the matrix T.

A configuration that **maximizes** the number of empty cells occurs when particles collapse into just three locations (leaving five cells truly particle-free). For example:

$$B : 3,\ E : 3,\ F : 1; \quad \text{all other cells: } 0.$$

Conversely, one that **minimizes** the number of empty cells (so there is only one particle-free cell and six occupied cells) can be obtained by spreading the particles as evenly as possible, e.g.:

$$E : 0,\ F : 2,\ A, B, C, D, G :\ \text{1 each.}$$

These two distributions illustrate the extremes of how a single transition can either clump particles into few cells or disperse them to cover almost all free cells.

(e) Particle weights and resampling distribution:
Assume the filter has two particles in A, three in B, one in E, and one in F, and we observe $e = \text{wall}$. From the sensor model:

$$P(e = \text{wall} \mid X = A) = 0.5, \quad P(e = \text{wall} \mid X = B) = 0.25, \quad P(e = \text{wall} \mid X = E) = 0, \quad P(e = \text{wall} \mid X = F)$$

Each particle in state s receives weight $w_s = P(e \mid s)$. The total unnormalized weight is

$$2 \cdot 0.5 + 3 \cdot 0.25 + 1 \cdot 0 + 1 \cdot 0.25 = 2.0.$$

Normalizing yields the resampling probabilities:

$$P_{\text{res}}(A) = \frac{2 \cdot 0.5}{2.0} = 0.50, \quad P_{\text{res}}(B) = \frac{3 \cdot 0.25}{2.0} = 0.375, P_{\text{res}}(E) = 0, \quad P_{\text{res}}(F) = \frac{1 \cdot 0.25}{2.0} = 0.125.$$

These weights define how particles are sampled in the next iteration.

Problem 3: Naïve Bayes (20 points)

Consider eight samples with $x_1, x_2 \in \{0, 1, 2\}$ and class $Y \in \{0, 1\}$:

Sample	x_1	x_2	y
1	2	0	0
2	2	2	0
3	1	0	0
4	1	2	0
5	2	1	1
6	1	1	1
7	0	2	1
8	0	2	1

(a) Parameter estimation. With $N_0 = N_1 = 4$ and Laplace $\alpha = 1$ over three values,

$$P(Y = 0) = P(Y = 1) = 0.5.$$

Smoothed feature CPTs:

$$P(x_1 \mid Y=0) : (0,1,2) = (1,3,3)/7, \quad P(x_2 \mid Y=0) : (0,1,2) = (3,1,3)/7,$$
$$P(x_1 \mid Y=1) : (0,1,2) = (3,2,2)/7, \quad P(x_2 \mid Y=1) : (0,1,2) = (1,3,3)/7.$$

(b) Prediction and accuracy. For each (x_1, x_2) compute

$$L_0 = P(x_1 \,|\, 0)P(x_2 \,|\, 0), \; L_1 = P(x_1 \,|\, 1)P(x_2 \,|\, 1).$$

Choosing the larger likelihood yields $\hat{y}$ correct for all eight samples, so the accuracy is 100%.

(c) E-step posteriors.

$$P(Y=0 \mid x) = \frac{L_0}{L_0 + L_1}, \qquad P(Y=1 \mid x) = 1 - P(Y=0 \mid x).$$

Sample	$P(Y=0 \mid x)$	$P(Y=1 \mid x)$
1	$(2,0)$	$(0.82, 0.18)$
2	$(2,2)$	$(0.60, 0.40)$
3	$(1,0)$	$(0.82, 0.18)$
4	$(1,2)$	$(0.60, 0.40)$
5	$(2,1)$	$(0.33, 0.67)$
6	$(1,1)$	$(0.33, 0.67)$
7	$(0,2)$	$(0.25, 0.75)$
8	$(0,2)$	$(0.25, 0.75)$

(d) M-step (no smoothing). Soft counts give $N_0 = 4.003$, $N_1 = 3.997$, so

$$P(Y) = (0.500,\, 0.500).$$

Updated CPTs:

$$P(x_1 \mid Y=0) : (0,1,2) = (0.125,\, 0.438,\, 0.438), \quad P(x_2 \mid Y=0) : (0,1,2) = (0.409,\, 0.167,\, 0.425),$$
$$P(x_1 \mid Y=1) : (0,1,2) = (0.375,\, 0.312,\, 0.312), \quad P(x_2 \mid Y=1) : (0,1,2) = (0.091,\, 0.334,\, 0.575).$$

Problem 4.5 – Analysis of the Experimental Results

1. **Mode 0 (forward algorithm)**— relationship between ϵ and localisation accuracy. Figure 1 shows the time–evolution of the average localisation error for five sensor–noise levels. All runs start from a near-uninformative prior, so the error is initially close to its theoretical maximum (≈ 2 for the Manhattan metric we are using). After a few time-steps the curves separate distinctly:

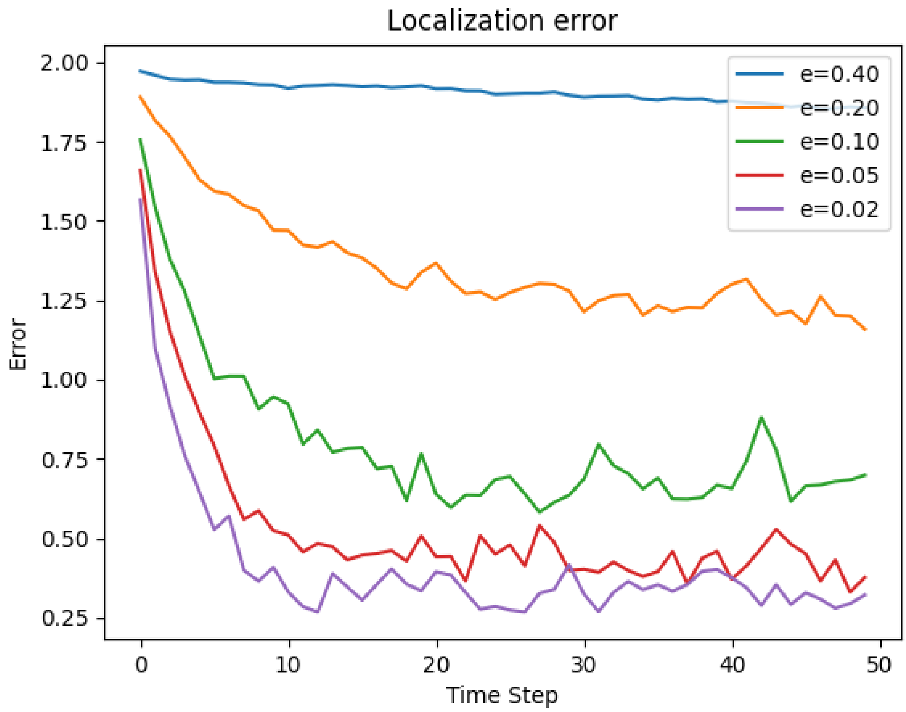

Figure 1: Average localisation error over 50 time-steps for the forward algorithm (Mode 0) under five sensor-noise levels.

- *Low noise* ($\epsilon = 0.02, 0.05$) rapidly drives the error below 0.4 and keeps it there. Because bits are flipped with probability ϵ, the likelihood function is sharply peaked and a single observation is already highly informative.
- *Moderate noise* ($\epsilon = 0.10$) still converges, but plateaus around 0.6–0.8. The forward algorithm can discount inconsistent states, yet the remaining ambiguity prevents convergence to a single cell.
- *High noise* ($\epsilon = 0.20, 0.40$) never leaves the high-error regime. When half the bits are wrong on average, contradictory evidence continually re-introduces states that were previously ruled out, so the posterior cannot sharpen.

The monotone ordering of the curves confirms the textbook result that, under exact inference, localisation performance degrades smoothly as sensor noise increases. The slight uptick of the $\epsilon = 0.10$ curve after $t \approx 30$ comes from occasional aliasing in the map: once the agent reaches a corridor with repeated geometry the observations become less discriminative, and the error rises.

2. **Mode 1 (particle filter, 200 particles)**— comparison to exact inference.

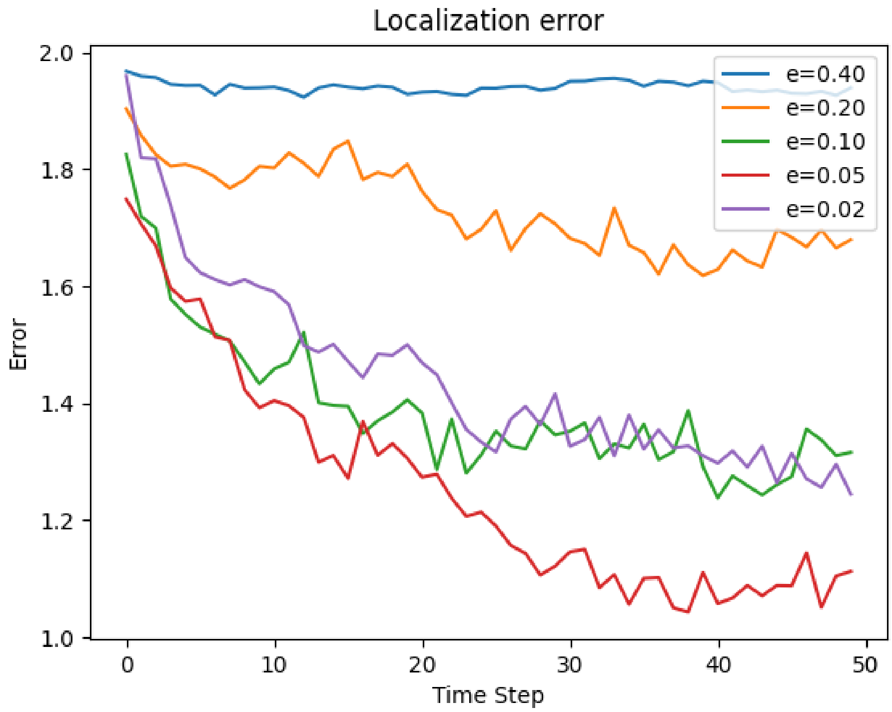

Figure 2: Average localisation error for the particle filter (Mode 1) using 200 particles and the same sensor-noise levels as in Fig. 1.

The particle filter reproduces the qualitative ranking of ϵ, yet every curve in Fig. 2 sits noticeably *above* its forward-algorithm counterpart. Two effects are visible:

- *Sampling variance.* With 200 particles the empirical distribution is only a coarse approximation of the posterior, especially during the first ten steps when weights are highly skewed. Hence the initial drop in error is slower than in the exact case.
- *Degeneracy and weight collapse.* After resampling, many duplicate particles traverse the *same* trajectory. When an unlikely observation occurs, large portions of the particle set receive near-zero weight, raising the mean error. This is why the $\epsilon = 0.02$ curve, although best at first, is later overtaken by $\epsilon = 0.05$ (around $t = 12$–15): a single inconsistent reading wiped out most of its particles.

In short, the particle filter is *consistent*—lower noise still yields lower error—but it is biased upward because of finite-sample effects.

3. **Mode 3 observations & explanation of tracking failures.**

Running the particle filter interactively reveals two characteristic failure modes:

(a) *Premature convergence to a wrong hypothesis.* If, by chance, the resampling step eliminates *all* particles near the true state, subsequent transitions are sampled only from the incorrect cluster. New evidence is then evaluated *locally* around that cluster, so the filter can remain *confident—and wrong—for many steps.*

(b) *Particle impoverishment in corridors.* In long corridors the motion model is nearly deterministic (move forward or stay), so the particles align in a thin band. When the agent makes a turn, there may be no particles adjacent to the actual corner cell, leading to a momentary explosion of error until diffusion repopulates the neighbourhood.

These failures are artefacts of a limited particle set; the exact forward algorithm never suffers from them because it maintains the full probability mass function.