(i) → finer discretization should represent reality better; But
(50 bins/dimension)

in 1K - training episodes plot of reward v/s no. of episodes we can clearly see opposite behaviour.
where 5 bins/dimension performed better than it.

- The number of total states will be way greater for 50 states per dimension than for 5 states per dimension.
- So, if we only have 1000 episodes, the agent will visit and update more faction of states of 5 bins/dim. than of 50 bins/dim..

- In case of 5 bins / dim., faster learning happens and gives better early performance ~~exactly what~~ ~~episodes demand~~.

- In case of 50 bins/dim., most states are never visited and thus learning is slow. 1K episodes are not enough to make Q-Table reliable.

Hence, conclusion is →
even though finer discretization is more accurate, it requires more data (higher no. of episodes). With limited no. of episodes, coarse ~~not so finer~~ discretization wins due to better states coverage.

(ii) → As explained in prev. answer (i);
The 50 bin/dim. discretization eventually will outperform the 5 bin/dim. discretization if no. of training episodes is increased.

That's what we can see in plot provided, that 50 bin/dim. discretization outperforms the 5 bin/dim. one after few thousand episodes and finally leads it with difference of large amount of reward.

Here;
The 5 bin/dim. discretization saturates after significant no. of episodes because agent visits most of the states in it and update Q-Table.
Whereas for the 50 bin/dim. discretization, agent ~~explores~~ explores many more states than it visited in prev(i).

Finally the conclusion we can draw is that if given sufficient episodes, finer discretization shows its advantage by learning more reliable/accurate policy.

(iii) Discretization is a useful method to handle continous state spaces as it helps convert continous values into discrete states so that Q-Table can be prepared.

However, discretization causes problems in edge cases: when divided into bins, very small change in value can move the state from one bin to another. So, even if two values are almost same, agent may take different actions for both because they are treated as different states.

(iv) → No, increasing the number of bins/dim. also increases the Q-Table size exponentially which can cause some problems such as; →

(a) most states will never be visited.

(b) learning becomes slow due to this and very high number of training episodes may be needed which isn't possible.

(c) also, saving such large Q-Table becomes impractical.