Intuition:

As a goal is to make not other blocks but the main block(2x2) move to the bottom opening, being able to slide the main block is the key. The block can move if the path is empty, in other words, if fewer blocks are blocking the main block's 'way', the main block is likely to get to the goal position easier(ideally in fewer steps). By the dimension of the main block, both of the empty squares are needed right infront of the block in that direction. So, it would be a nicer guide if the heuristic function cares about whether moving towards is possible right away, without spending costs on other less valuable blocks.

For a simpler computation, we will focus on sliding downwards especially as the boad is longer vertically, and the bottom opening is in the middle. (Note also the positions of the empty squares and the main block are stored as attributes, so there's no need for search.) By the dimension of the main block, some of its pieces(if we think of the block as a 4 single piece) are inevitably on at least one of the two middle columns of the board. In other words, it is the path that the main block that needs to take at some point. So having empty squares downwards is likely a way to essentially fast forward of the path to the goal. Utilizing the idea of the manhattan distance heuristic, denoted by $h_m(n)$, as it is a line along theminimum desired path ignoring other blocks, we can simply add 0 or 1 for each case we have 2 empty squares or not, as bringing 2 of empty squares infront is key to reduce less valuable steps.

An advanced heuristic function $h(n)$ is defined as:

$$h(n) = \begin{cases} h_m(n) & \text{if there are 2 empty squares right below the main block} \\ h_m(n) + 1 & \text{otherwise} \end{cases}$$

on state n.

$h(n)$ **<u>is admissible.</u>** By contradiction, consider the negation; there is some state n_0 that makes $0 \leq h(n_0) \leq h^*(n_0)$ false, i.e.,

$0 > h(n_0)$or $h(n_0) > h^*(n_0)$. We will show you that both are not possible for any n_0.

(1) $0 > h(n_0)$ is impossible as $h(n_0)$ is essentially about adding natural number or not to the Manhattan distance which we know to be greater or equal to 0.

(2) $h(n_0) > h^*(n_0)$ is impossible, i.e., $h(n_0) \leq h^*(n_0)$ for all states.

Considering when $h^*(n_0)$ =0, n_0 is the goal state. Then, there can't be an empty square below the main block as the goal position is located at the bottom of the board. Therefore, $h_m(n_0) + 0 = 0 = h^*(n_0)$.

Next, $h^*(n_0)$ =1 is when the main block can reach a goal position by 1 move. To do that, the main block must be located in one of the bottom left, the bottom right, and the middle of the second last row. Plus, it must have two empty squares beside or below correspondingly to make it in one move without moving other blocks around. However, since we only count for the empty squares that are below the main block for simple computation, nothing is added. Therefore, $h_m(n_0) + 0 = h^*(n_0)$.

Because there are only two empty squares, the way of the main block possibly heading towards is inevitably blocked by the non-empty squares if the main block is more than one move farther from the goal, or in other words, far by more than 1 Manhattan distance from the goal, or when $h^*(n_0) \geq 2$.

When blocked, the non-empty blocks on the 'way' should be 'cleaned up' before the main block's move. This cleaning up takes at least one step. When not blocked, the main block can move, but after that one move, both empty squares are again behind the main block, which will cost 2 or more steps. At this point, without cleaning up again, it can only undo its move, which is not optimal at all. In other words, every move of the main block requires cleaning up as next steps no matter having an empty square on its path initially or not. Therefore, we can say for state n_0 where it needs 2 or more moves to the goal, after every move along the cheapest paths, that is, excluding the undo-movement, the 'cleaning up' is done to perform its next move. So as we at least need two main block's move as well, we know

$\forall h^*(n_0) \geq 2,\ h_m(n_0) + 1 < h*(n_0) \Rightarrow h(n) \leq h*(n)$, that $h(n)$ never overestimate the cost of the cheapest path from state n to a goal state.

$h(n)$ **<u>dominates the Manhattan distance heuristic.</u>**

We know by the definition of $h(n)$ that is about adding a positive integer or not to the Manhattan distance heuristic, $h(n)$ can only be greater than the Manhattan distance heuristic. That means, it can never be smaller in any case, and is strictly greater in at least one case(eg. The main block located at the middle of the second last row and has an empty square between the goal position and its current state. Note it is still $h(n) \leq h*(n)$, being admissible). Therefore, $h(n)$ dominates the Manhattan distance heuristic.