The algorithmic task of computing the Hamming distance between a given pattern of length m and each location in a text of length n is one of the most fundamental algorithmic tasks in string algorithms. Unfortunately, there is evidence that for a given text and pattern, one cannot compute the exact Hamming distance for all locations in the text in time which is polynomially less than \( o(n \sqrt{m}) \). Nevertheless, Karloff showed that if one is willing to suffer a \( 1+\epsilon \) approximation, then it is possible to solve the problem with high probability in \( O(n / \epsilon^2) \) time.

Due to related lower bounds for computing the Hamming distance of two strings in the one-way communication complexity model, it is strongly believed that obtaining an algorithm for solving the approximation version cannot be done much faster as a function of \( 1 / \epsilon \). We will show that this belief is false by introducing a new \( O(n / \epsilon) \) time algorithm that succeeds with high probability.

The main idea behind our algorithm, which is common in sparse recovery problems, is to reduce the variance of a specific randomized experiment by (approximately) separating heavy hitters from non-heavy hitters. However, while known sparse recovery techniques work very well on vectors, they do not seem to apply here, where we are dealing with mismatches between pairs of characters. We introduce two main algorithmic ingredients. The first is a new sparse recovery method that applies for pair inputs (such as in our setting). The second is a new construction of hash/projection functions, which allows to count the number of projections that induce mismatches between two characters exponentially faster than brute force. We expect that these algorithmic techniques will be of independent interest.