ML as a component of exact algorithms

1. What does it mean to have an ML-accelerated algorithm that is “exact”?
   Discuss the distinction between exact algorithms, interpretable algorithms, and algorithms that allow error propagation.

2. What are the differences between in-principle and in-practice exactness? Are they important?

3. In what applications (both in LQFT and drawing parallels to other ares in physics) is it important to guarantee exactness in ML-accelerated algorithms, and where is it unnecessary, impossible, or worth sacrificing?