We thank all reviewers for the time spent on their comprehensive reviews and we are happy that they enjoyed reading our paper. We are particularly glad to see reviewers unanimously agreeing that our work is a well-motivated and significant contribution to the NeurIPS community, providing a needed solution for a problem with immediate applications across a wide range of fields. In this rebuttal, we respond to each reviewer's minor comments individually, as there are no comments shared across reviews.

Reviewer 1

Thank you for the extremely positive review. We will fix the one typo reported.

Reviewer 2

Thank you for the strongly positive review, arguing for acceptance based on strong performance across our comprehensive and thorough experimental section. In light of your comments around our definitions of λ_m and λ_g, we will endeavour to improve clarity in the final version, emphasising that λ_m and λ_g are scalar quantities representing our kernel's two hyper-parameters. These parameters tune the contribution of highly non-contiguous sub-sequences in our kernel calculations. We will also address your comments about combining the description of our proposed framework into a single location.

Reviewer 3

Thank you for the predominately positive review, stressing the novelty and importance of our work to the NeurIPS community. We regret that minor clarity issues around our use of genetic algorithms prevented you from having as pleasant reading experience as our other reviewers.

No general guarantees exist for the convergence of genetic algorithms over string spaces and their derivation would be a substantial contribution in its own right. However, genetic algorithms were designed with string optimisation problems in mind and are widely used to explore string spaces (see Whitley et al. [1994]). Note that exact convergence of our genetic algorithm is not a requirement for effective BO as exact acquisition maximisation is rarely achieved even when performing BO over continuous spaces. All that is typically required is the identification of a string that scores reasonably highly according to our acquisition function.

We respond to your numbered comments as follows:

1. Rubanova et al. [2020] also consider Bayesian optimisation over strings. However, their approach, using ensembles of neural networks and generative models for acquisition function maximisation, has a smaller scope than our work, seeming suitable for only shorter strings of up to 100 characters and unconstrained string spaces. We will discuss this contemporaneous work (published at the start of this month) in our final version.
2. In Figure 5, we consider a gene optimisation problem over very long strings of 3672 base pairs and we use our locally-constrained genetic algorithm to explore this space of strings. It is unclear how a random forest (typically a classification or regression model) could be used to provide optimisation over discrete (never-mind syntactically constrained) spaces and this is beyond the scope of our work.
3. There is a stream of work combining convolution kernels and GPs for NLP applications (such as the cited Beck and Cohn [2017]), including kernels for modelling trees and graphs. Our method could definitely be extended to such kernels, providing efficient optimisation over additional types of discrete structures. As we mention in our Discussion section, we plan to investigate these avenues in future work.
4. We have presented numerical results for the efficiency of ours (and competitors) BO routines in Figure 4 of the main paper and in Figures 11-17 of our Appendix. We will better signpost this in the final version.

Reviewer 4

Thank you for the extremely positive review and helpful minor comments that we will address to improve clarity in the final manuscript. We now deal with your concern that our kernel parameter λ_m is redundant after kernel normalisation. As our kernel (defined in line 105) contains the sum over all sub-sequences containing up to n characters, each kernel evaluation is a linear combination of λ_m raised to each power in $[2, .., 2n]$. As the coefficients of this linear combination depends on the two considered strings, a dependence on λ_m remains even after the normalisation of line 111. The choice of both λ_m and λ_g is important to the efficacy of the kernel and this is explicitly demonstrated for molecule prediction in Figure 18. In the final version, this figure will be brought into the main paper alongside further clarification of the role of the kernel parameters.