Generel comments:

- Summary: Layout design and wake steering (achieved through yaw control) is combined in a co-design process that does not exceed the computational cost of a traditional layout optimization by determining yaw angles from the layout.
- The work is well-placed in context to related literature and clearly motivated.
- The reported 0.8% increase in AEP is indeed a very interesting result – but how generally applicable is it? In this paper it is based on one comparison. It would be interesting to have further comments on this and the limitations of their work from the authors.
- All in all a useful, very well-written and interesting study.

Specific comments:

Section 1. Introduction

- p. 3. Fig. 1: As a reader, it would be helpful with additional information in order to better relate to the stated computation time for an optimization. What are the computational resources used (e.g., amount and type of CPUs)? How well are these optimization problems converged (in SNOPT that would be the Major Optimality Tolerance)? How are the gradients computed (further down on p. 9 finite difference is mentioned but that is for the SciPy setup)? The information cannot fit into a single figure caption so one idea could also be to fit them into the Section 2 on the Geometric Yaw Relationship where SNOPT is mentioned. Still, without more information it is difficult for a reader to relate to the reported computation time.
- p. 3. Fig. 1: Why are there three dots on the curve?
- p. 3: 'Revolutionize' is as a word brought up three times in this paper. Abstract, here, and finally in the Conclusions (where it says 'can revolutionize' and not will revolutionize). Indeed, the method is clearly an improvement – but the 0.8% presented further down is based on one comparison. One could perhaps modify the wording slightly here – or to build confidence of the significance for the findings presented in this study - somewhere include comments of how generally applicable these results are. What about some of the factors not taking into considerations (comparing orders of magnitude in effect)?

Section 2. Geometric Yaw Relationship

- p. 4: See comment on Fig 1 above. It would be interesting to have more information on the optimizations.
- p. 4 On the geometric yaw relationship:
  It is based on the (very illustrative) plot in Fig. 2C. Still, the resulting yaw relationship in Fig 2.D seems 'manually' constructed (i.e., "defined through observation of the pattern"). Why not simply interpolate using e.g., a piecewise linear model or perhaps a curvature-minimizing interpolant in 2D? Could the authors' reasoning be, that these potential improvements would only lead to minimal extra gain? This could be further elaborated.

Section 3: Results

- p. 6: suggestion: SciPy instead of scipy.
- p. 6: Spelling typo: SLSQP gradient-based optim[i]zer
- p. 6: What was the reason for changing optimization setup from SNOPT to SciPy?
- p. 6: Was it necessary to add the final continuous yaw optimization in the second example where the formulation leveraging geometric yaw relationship is tested? It would seem, that the ideal incorporation of the geometric yaw in the optimization procedure would not need a final yaw optimization at the very end. How much improvement is gained from this extra optimization? Could this extra step be avoided somehow (e.g., through a more accurate

geometric yaw relationship based on interpolation)? More comments on this would be interesting.

- p. 9: Spelling typo: 0.8 % higher AEP [that] → 0.8% higher AEP [than].
- p. 9: When comparing the final layout result for the two optimization strategies (seen in Fig. 4C-D) it would be interesting to also see the starting position of the two layouts. This is relevant since the final part of the Results section speaks to the disparate final layouts obtained with the two methods. This is indeed a clear tendency also seen from the illustrative 1-D example. Still, a contributing factor not mentioned could also be, that the two strategies started from different baseline layouts (each method could start from any of the 50 candidate layouts). This could be further clarified in the text.
- p. 9: Further information on the two final optimizations being compared would be of interest. How many function calls for each optimization? How well were the optimization problems converged? Comments on the computational resources, computation time used, etc.

Section 4: Conclusions

- p. 9: New information concerning number of function calls is introduced for the first time in this section. Ideally, this point should be introduced and clarified further already in the Results section before being mentioned in the conclusion. Other than that it is a nice, succinct conclusion.