Reply to referee comment RC2

We thank the referee for carefully reviewing the manuscript and for the valuable suggestions and comments. The reviewer's comments are highlighted in blue and the authors' response in black. *Changes to the manuscript are displayed in italic font*.

The manuscript introduces CrystalTrace, a Monte Carlo raytracing algorithm designed for simulating radiative transfer in cirrus clouds containing ice crystals with both random and preferred orientations in the visible spectral range. It runs as a standalone tool to compute scattering phase functions and relative intensities for hexagonal prisms of varying sizes and aspect ratios, while being integrated into the MYSTIC solver within the libRadtran library to enable multiple-scattering simulations of absolute radiances in complex atmospheric scenes. The algorithm uses geometric optics, accounts for crystal orientations via Euler angles and Gaussian distributions for tilt, and validates its single-scattering properties against established methods like IGOM, showing good agreement except in forward scattering regions where diffraction is not included. CrystalTrace extends MYSTIC's capabilities by computing single-scattering properties online, reducing memory needs and eliminating precomputed look-up tables, and demonstrates its utility through simulations of atmospheric optical phenomena such as sundogs, tangent arcs, and Parry arcs. Overall, the tool addresses a gap in efficient forward modeling for oriented ice crystals, facilitating improved retrievals of cirrus properties from ground-based, airborne, and satellite observations.

Overall, this is an excellent study: well-organized, clearly written, and easy to follow. The development of CrystalTrace represents a valuable advancement in modeling radiative transfer for oriented ice crystals, addressing a key limitation in traditional approaches that assume random orientations. As an integrated component of the widely used libRadtran package, it holds significant potential for applications in remote sensing, and atmospheric optics, particularly for simulating optical phenomena like halos and improving cirrus cloud retrievals. I recommend that the manuscript be accepted for publication after the authors address the following minor clarifications and revisions.

Minor comments:

Comment 1: Currently, CrystalTrace does not account for diffraction, which is a known limitation of pure geometric optics (GO) methods for forward scattering regions. Can the authors provide more details on how diffraction effects are handled (or approximated) in the radiative transfer simulations described in Sections 2.3 and 2.4? For instance, did you apply forward peak truncation to the phase functions in the GO and IGOM simulations to mitigate delta-function transmission issues?
Response 1: CrystalTrace is a purely geometric optics model, and diffraction is not accounted for. In the radiative transfer simulations described in Sections 2.3 and 2.4, forward diffraction peak is represented by rays that continue propagating forward through the crystal. Because the model does not produce a delta function like forward peak in the phase function, no forward peak truncation or delta scaling was applied.

Comment 2: In Figure 7, there is a small peak around the 170° scattering angle in both the GO and CrystalTrace phase functions, which appears to be absent in the YANG phase function. Could the authors explain the potential cause of this difference?
Response 2: CrystalTrace inherited this small peak for the original GO raytracing solution, which was shown in Macke et al. 1996 and most likely stems from high-order internal reflection rays.

Comment 3: How straightforward is it for users to implement new crystal shapes in CrystalTrace?

For example, rare halos like the 9° halo have been linked to irregular ice crystal shapes such as droxtals (Zhang et al., 2004, Appl. Opt., 43, 2490-2501). Would adding such a non-hexagonal or complex geometry require significant code modifications, or is the framework flexible enough for users to extend it with minimal effort? Providing a brief example or guidance in the manuscript could enhance its accessibility.

Response 3: In principle, CrystalTrace can support the same types of hexagonal ice crystal geometries that Macke 1994's raytracing code has been used with. We have only tested hexagonal prisms with different aspect ratios, but the framework can be extended to other geometries. Users can generate new shapes by modifying the Python script in the GitHub repository that creates the geometry input files.
(https://github.com/lforster/CrystalTrace/blob/main/crystal_geometry/generate_crystal_file.py). We would like to note that for small, randomly oriented particles such as droxtals, optical properties from electromagnetic scattering methods, such as Yang et al. (2013) are preferred. The strength of CrystalTrace comes in for simulations of larger crystals with preferred orientations. We modified the abstract to clarify this.
Page 1, line 7: "The current version of CrystalTrace is implemented for the visible spectral range and supports individual hexagonal prisms with different sizes and aspect ratios."

Comment 4: Can the authors clarify in which version of libRadtran CrystalTrace will be (or has been) incorporated? As of the latest public release (version 2.0.6, December 2024), there is no mention of it on the official website, so specifying the target version or branch (e.g., a development or upcoming release) would help potential users.

Response 4: Thank you for this suggestion. CrystalTrace is part of libRadtran version 2.0.6 and includes an example as test case but has not yet been documented in the user manual. We provide additional documentation and examples in a GitHub repository. We modified the code and data availability statement:
Page 15, line 261: "CrystalTrace is distributed as part of the libRadtran radiative transfer package (version 2.0.6 and later). Supplementary materials, including detailed documentation, example input files, and plotting scripts used in this work, are available on GitHub (https://github.com/lforster/CrystalTrace) and have been preserved on Zenodo (DOI: 10.5281/zenodo.17679332)."