Authors' Response to Emily de Jong's Comment RC#1

on Manuscript "CLEO: The Numerical Methods of a New Superdroplet Model including a Droplet Breakup Algorithm" , now renamed "Cleo: The Numerical Methods of a New Superdroplet Model including a Droplet Breakup Algorithm (v0.52.0)" by Clara J.A. Bayley et al. (2026)

Original Title	CLEO: The Numerical Methods of a New Superdroplet Model including a Droplet Breakup Algorithm
New Title	Cleo: The Numerical Methods of a New Superdroplet Model including a Droplet Breakup Algorithm (v0.52.0)
Author(s)	Clara J.A. Bayley et al.
MS No.	egusphere-2025-4399
MS type	Model description paper

We thank Emily for her constructive criticism of the manuscript and for seeing the purpose and value of our work. We've addressed each of the points she raised with amendments and added content, especially so with regard to balancing the introduction, to verifying the more novel aspects of Cleo's numerical methods, and to discussing Cleo's parametrisations and coupling.

Please note the updates to the dataset for this paper (https://doi.org/doi:10.17617/3.SDN0NX) are currently viewable from this link whilst they await publishing:
https://edmond.mpg.de/previewurl.xhtml?token=6ed04ec0-fd10-4ce4-a508-09280f4f2a14.

Major Comments

1. *The abstract and introduction include strong and potentially misleading descriptions of traditional microphysics methods that would benefit from a more nuanced presentation. Specifically:*

   - *The abstract mentions "Eulerian models" but should be careful to distinguish microphysics parameterizations from the entire class of "Eulerian" models; all ESMs are Eulerian in their treatment of momentum, after all.*

     We now specify in the abstract (P1 L1) "conventional Eulerian**, bulk and bin,** models". Also elsewhere we now specify "Eulerian, bulk and bin, models", "Eulerian models for cloud microphysics", "Eulerian microphysics models", or "Eulerian cloud microphysics models" whenever we use the word "Eulerian" in referring to bulk or bin models for cloud microphysics.

   - *Bold claims like "their underlying assumptions are inconsistent with even the most basic kinematic processes" need elaboration or tempering*:

     We agree this paragraph was too bold and for this particularly problematic sentence we think that's primarily because it was too vague. We've therefore amended it to repeat that we are specifically referring to bulk schemes and also to make it more specific. It now reads (P2 L27-31) "Varying definitions of condensate size distributions in bulk schemes are also problematic because they alter shallow-cloud fraction and albedo, even in non-precipitating clouds, and their underlying assumptions, regarding the functional form of each distribution and distinguishing between "rain" and "cloud" condensates, are known to be inconsistent with observations and with bin scheme models of cloud microphysical processes (e.g. Igel and van den Heever, 2017a, b; Igel, 2019; Igel et al., 2022).".

     In the rest of the paragraph we've added more citations to support certain statements: (P2 L31-34) "When Large-Eddy Simulations (LES) and observations are compared, differing precipitation patterns are frequently attributed to differences in microphysics schemes (e.g. **Ackerman et al., 2009**; King et al., 2015; Schulz and Stevens, 2023), and bin schemes have been shown to be just as disparate as bulk ones (vanZanten et al., 2011; **Hu and Igel, 2023; Hill et al., 2023**).", and (P2 L37-40) "Differences inevitably arise from gaps in our knowledge of cloud microphysics, however, the differences highlighted above are not caused by knowledge gaps, but rather by "model uncertainty", uncertainty inherent to the way that conventional Eulerian cloud microphysics models represent what we already understand **(Grabowski et al., 2019; Morrison et al., 2020)**.".

We have also softened the language (P2 L33) saying "bin schemes are as disparate as bulk ones" to "bin schemes have been shown to be just as disparate as bulk ones"; and later on in the introduction (P2 L71-73) we replaced "[...] SDM is clear and direct, meaning that variations between SDMs hold a strong relation to differences in our understanding of the underlying physics." to "[...] SDM is clearer and more direct than in Eulerian microphysics models, meaning that variations between SDMs hold a stronger relation to differences in our understanding of the underlying physics".

- *Claims that SDM converges toward the Smoluchowski equation while bulk or bin schemes do not are needs elaboration. Consider that linear sampling in the SDM would lead to slower than expected convergence, wherease bin schemes provide a direct finite element approximation to the Smoluchowski equation.*:

  We sincerely apologise that we had completely incorrectly expressed this. We now correctly state (P3 L62-70): "[...] the mean of many simulations of its collision-coalescence algorithm approaches the solution to the Smoluchowski equation (Smoluchowski, 1916). This is in stark contrast to bulk schemes, where increasing refinement does not converge to a generic solution, and also to bin schemes which model collisions by solving the Smoluchowski equation directly. Bulk schemes therefore suffer from the limitations of that equation; namely its mean-field approximation assumes that the locations of differently sized droplets are uncorrelated (Gillespie, 1972), even though this is only variably true (Dziekan and Pawlowska, 2017; Morrison et al., 2020), and prevents modelling "lucky drops" which may be crucial for precipitation onset (Telford, 1955; Kostinski and Shaw, 2005; Dziekan and Pawlowska, 2017; Morrison et al., 2020). In SDM, the stochastic collision-coalescence algorithm does not make these assumptions and can represent lucky drops (Dziekan and Pawlowska, 2017; Li et al., 2022).".

- *The introduction should recognize that the rationale for developing bulk and bin microphysics largely stems from computational limitations that necessitate approximations; the SDM is still infeasible for global simulation, thus we will continue to rely on bulk microphysics. In fact, many non-Lagrangian approaches are based on clever mathematical approximations and have enabled broad advances in atmospheric science based on necessary complexity tradeoffs. Lagrangian microphysics is itself not free of assumptions or limitations, nor differences between implementations (see Hill 2023) and these should be reasonably discussed in similar level of detail.*:

  We have restructured the first paragraph which introduces SDM in the introduction (P2 L41- P3 L56) and added new text (P2 L46-52) stating that "The major limitation of SDM is its computational cost, both in terms of its memory consumption and its number of computations, which presumably is the reason that LES using SDM have so far been limited to domains $\mathcal{O}(10\,\mathrm{km})$ or smaller (to the best of our knowledge, the largest LES with SDM to-date are Sato et al. (2017); Chandrakar et al. (2021); Matsushima et al. (2023) and Yin et al. (2024)). Historically, Lagrangian particle-based microphysics schemes have struggled computationally (e.g. Jacobson, 2005; Simmel and Wurzler, 2006) whilst bin and bulk schemes have advanced not only cloud process understanding but also numerical weather and climate prediction in ways that would not otherwise have been possible, by working within the computational constraints of the time (Khain et al., 2015; Morrison et al., 2020)". In the paragraph thereafter we have now added that (of course!) SDM makes modelling assumptions and hence has limitations (P4 L57-61): "**Whilst SDM is not free of modelling assumptions and hence still suffers from its own sources of model uncertainty (Hill et al., 2023)**, SDM makes no assumptions about condensate categories or size distributions unlike bulk schemes, and it does not suffer from same intrinsic problems as bin schemes, such as numerical diffusion and the "curse of dimensionality" (Grabowski et al., 2019).".

2. *Some of the new additions to CLEO (ventilation, breakup) have not been rigorously validated in the included experiments.*

   - *Ventilation was not validated for expected qualitative behavior (cooling, evaporation of rain) in any of the example cases. Please show an instance where the ventilation effect can be qualitatively observed.*

     We have added another set of adiabatic parcel simulations to Section 6.1 ("Verification of Condensation / Evaporation") and we also now point the reader to more realistic simulations which used Cleo and showed ventilation's effect on droplet size distributions and evaporation in the sub-cloud layer beneath shallow cumuli (Niebaum et al., 2025). The following extract from Section 6.1 contains our new analysis (P17 L400-421):

     "To verify qualitatively the effect of ventilation on droplet growth, we ran a second set of adiabatic parcel simulations (Figure 8). In these simulations we adapted the setup above to demonstrate the behaviour of large droplets falling from a cloud through subsaturated air. We initialised the parcels with wet droplets in 900 hPa pressure, 290 K, and 98.0% relative humidity, and then let them descend with a constant fall speed until they reached a pressure of 1032 hPa. The wet droplets all had the same dry radius of 0.1 μm but we ran the parcel model with three different initial wet droplet radii:

0.1 mm, 1 mm, and 5 mm, and droplet number concentrations of 0.625 $\mathrm{cm^{-3}}$, $6.25 \times 10^{-3}\,\mathrm{cm^{-3}}$, and $5 \times 10^{-6}\,\mathrm{cm^{-3}}$ respectively, so that the mass of condensed water at the start of each simulation was 2.61 $\mathrm{g\,m^{-3}}$. The fall speed of each parcel was set to approximately the fall speed of the initial droplets, at 1 $\mathrm{m\,s^{-1}}$, 5 $\mathrm{m\,s^{-1}}$, and 10 $\mathrm{m\,s^{-1}}$, respectively. For each parcel, we ran the simulation once with the ventilation factor as in Equation 3 and once without, in which $f_v = 1.0$.

Figure 8 shows how the ventilation factor increases the evaporation of droplets and consequently affects the parcel's relative humidity. For each parcel the droplet radii decreases faster with ventilation accounted for than without. The droplets which are initially 0.1 mm-sized evaporate completely before the end of the simulation, causing substantial moistening of the parcel and the final relative humidity and droplets' size to be the same regardless of whether ventilation was accounted for (Figure 8a). However, the relative humidity is significantly higher and the droplets are significantly smaller earlier in the simulation (higher in the atmosphere) with ventilation. Compared to the parcels with 0.1 mm-sized droplets, the parcels with larger droplets are substantially drier, reaching 50-60% relative humidity by the end of the simulations compared to 90%. This is because the larger droplets have a much lower rate of evaporation and much faster fall speed, which results in much less moistening of the parcel. Also because the evaporation rate is so low, the relative humidity of the parcels increases only by up to a few percent when the ventilation effect is accounted for. However, since the rate of evaporation is up to an order of magnitude larger, the final radii of the droplets are $10 - 100\,\mu$m smaller. In more realistic simulations also using Cleo, this effect has been shown to be even larger and have a significant impact on evaporation in the sub-cloud layer beneath shallow cumuli (Niebaum et al., 2025).".

- *The choice to eschew fragment size distributions altogether is a break from the norm, and warrants further discussion and validation. I recommend including a comparison of the stationary PSD that results from this choice of fragment size with those of traditional implementations (Straub 2010, McFarquhar) to reveal any systematic biases that that your implementation may suffer, such as overproduction of cloud-size fragments.*

  In Section 6.2 ("Verification of Collisions") we have added a second set of box-model simulations using our treatment of collisions which includes breakup. The stationary droplet size distributions from these simulations are now plotted in Figure 10 and are compared with the stationary droplet size distribution from Straub et al. (2010). This is the same comparison as done by de Jong et al. (2023a, Sec. 4.1.2, Figure 9) except that we do not sample the parametrised fragment distribution from Straub et al. (2010), but rather consider different parametrisations for the number of fragments that can be produced by collisional breakup events. The following extract from Section 6.2 contains our new analysis (P18 L440-465):

  "To demonstrate the behaviour of our collisional breakup algorithm in comparison to an approach which uses a parametrised fragment size distribution, we ran a second set of box-model simulations. In these simulations, we compare the stationary droplet size distribution to that reported by Straub et al. (2010, Sec. 5, Figure 10). This is the same comparison conducted by de Jong et al. (2023a, Sec. 4.1.2, Figure 9) except that our algorithm does not sample the parametrised fragment size distribution, but rather prescribes a certain number of fragments depending, in general, on the superdroplets which collided (Section 3.2). The simulations follow the same setup as Straub et al. (2010), using an initially exponential distribution as expressed by Marshall and Palmer (1948), whereby $p(R) = N_{\mathrm{mp}} e^{-2\lambda_{\mathrm{mp}} R}$ and $N_{\mathrm{mp}} = 8 \times 10^6\,\mathrm{m^{-4}}$, $\lambda_{\mathrm{mp}} = 4.1 \times 10^3 R_{\mathrm{mp}}^{-0.21}\,\mathrm{m^{-1}}$, and $R_{\mathrm{mp}} = 54\,\mathrm{mm\,h^{-1}}$. We use 8192 superdroplets to sample the initial droplet size distribution evenly in log-radius-space bins between $0.5\,\mu$m and 4 mm (Figure 10a). To make the results comparable to Straub et al. (2010), in these simulations we do not use the parametrisation from Testik (2009) to include possibility of rebound, but rather we directly determine whether a collision results in either coalescence or breakup by comparing a random number with the Straub et al. (2010) collision efficiency, according to Equation 14.

  Figure 10b shows how the stationary droplet size distribution in Cleo with different parametrisations for the number of fragments compares to the distribution from Straub et al. (2010) with a parametrised fragment size distribution. Using a fixed number of fragments in Cleo, one can sample different parts of the stationary distribution and, as expected, increasing the number of fragments produces a narrower distribution of smaller droplets. Using the parametrisation for the number of fragments given by Schlottke et al. (2010) (as described in Appendix A), the stationary distribution appears qualitatively similar to the cases with a fixed number of fragments, suggesting that the parametrisation tends to produce a similar number of fragments for all collisions which result in breakup in the steady state. In all cases, the shape of the droplet size distribution is narrower than that given by Straub et al. (2010), concentrating more droplets in a smaller size range. A different parametrisation for the number of fragments, for example one which depends on the size of the colliding droplets directly, may show a greater fidelity to the shape of the stationary distribution according to Straub et al. (2010). This example therefore demonstrates how Cleo's overly simple approach to fragment sizes does not seek fully to reproduce measured or parametrised fragment distributions, but rather that the approach can be used to probe upper and lower bounds for the effect breakup could have on a droplet size distribution, for example by using a very high or a very low fixed number of fragments as done by Niebaum et al. (2025).".

Additionally we have now added a supplement to the manuscript where in Section 1 we include box-model simulations comparable to de Jong et al. (2023a, Sec. 4.1.1, Fig. 7a). The following extract contains this new analysis (Supplement Section 1; P1 L1-16):

"Figure 1 shows the same setup as for the test of SDM collisions as in Shima et al. (2009, Sec. 5.1.4, Figure 2), which is also used to produce Figure 9 in Section 6.2 of the main manuscript. However, here we show the mass density distribution after 120 s for simulations comparable to de Jong et al. (2023a, Sec. 4.1.1, Fig. 7a). Specifically, the simulations use 8192 superdroplets and the hydrodynamic collision kernel with collision efficiencies from Long (1974). Collisions either result in only coalescence (as per the original treatment of collisions; Shima et al., 2009), or use our extended treatment. In the extended treatment for these simulations rebound is excluded and a random number is compared to a fixed coalescence efficiency, $E_c = 0.95$, to determine if breakup or coalescence occurs. The number of fragments produced in the event of breakup is also fixed such that $N_{\mathrm{frag}} = 4.0$, 16.0 or 64.0.

The results are qualitatively the same as de Jong et al. (2023a) but the droplets are generally larger for the same fixed number of fragments. As the number of fragments increases, the mean droplet size decreases and the distribution broadens. As explained by de Jong et al. (2023), this broadening of the size distribution is generally expected because the small droplets produced by collisional breakup have a wide range of collision rates. For the same N_{frag}, the mean of the distribution is, however, noticeably larger than de Jong et al. (2023), possibly because we do not allow for multiple collisional breakup events in a single time-step and therefore our algorithm produces fewer fragments with larger sizes for cases in which PySDM instead iteratively produces N_{frag} fragments.".

In conducting these validations we also noticed that the rescaling of the random number we had used to determine if the outcome of a collision resulted in coalescence, breakup, or rebound was incorrect. We therefore amended our code to allow either for rescaling the random number (as we are currently experimenting with) or for drawing a new random number instead. In the manuscript we have therefore adjusted the text describing how we determine the outcome of a collision (P9 L224-227): "In this example and similarly to de Jong et al. (2023a), we draw a new random number ϕ'_α and we compare ϕ'_α with E_c to decide if the collision results in coalescence or the alternative. In general however, the outcome of a collision in Cleo does not need to depend on a second random number, it can instead be calculated from the superdroplet attributes or by rescaling the original random number."; and in P12 L278-281 where we point out the differences between Cleo's and pySDM's collisional breakup algorithm we now state: "Our algorithm is similar to that of pySDM (de Jong et al., 2023a), but with two adjustments. Firstly, although the step to determine if a collision does or does not occur is always probabilistic (Section 3), in general whether a collision then results in rebound, coalescence, or breakup in our algorithm does not have to depend on drawing a new random number, whereas in de Jong et al. (2023a) an additional probabilistic step is always used.".

Due to this change in the code, in Section 6.2 ("Verification of Collisions") we have also replotted Figure 9 so that the simulations including breakup use a second random number to determine the outcome of collisions. Our results and conclusions remain unchanged, except that the stationary distribution including breakup centres on 550 μm instead of 650 μm. In the text we therefore now state "In Figure 9b, up to about 600 s the evolution of the droplet size distribution is comparable to the coalescence-only case, but the subsequent growth of droplets is much delayed and converges to a stationary distribution centred at **550** μm, rather than 2400 μm in the coalescence-only case; and qualitatively the same behaviour is seen in Figure 9c too.".

- *Figure 10: Including a plot of CLEO's predictions of precipitation rate and timing would further be extremely useful for comparison with other SDMs in the Hill 2023 study (and would back up the claim in L350-351).*

  Corresponding with the same plots in the first column of Figure 3 in Hill et al. (2023), we've now included Figure 13 in our manuscript. It plots the liquid water path, surface precipitation, cloud droplet number concentration, mean volume diameter and its standard deviation in the same way as Hill et al. (2023) but with a smaller ensemble size and without any smoothing they might have used. Correspondingly in the main text we have replaced "(not shown)" with "(Figure 13)" in P19 L495-497: "The mean precipitation onset is around 30min, in agreement with the range of SDMs in Hill et al. (2023), and likewise the evolution of the liquid water path, cloud droplet number concentration, mean volume diameter and its standard deviation also lie within the SDMs' spread **(Figure 13)**.".

- *Seeing a verification example where the terminal velocity is included in addition to the passive tracer flow (Figure 9) would be interesting.*

  We have now added a supplement to this manuscript where in Section 2 we include an example in which the tracers have a terminal velocity. Furthermore there is an example in Cleo's repository called "constthermo2d" which users can play around with (for details on how to do this see: https://yoctoyotta1024.github.io/CLEO/usage/examples/examples.html). We chose to put these simulations in a supplement because droplet terminal velocities are already demonstrated in Figure 12 (previously Figure 10) and hence we do not believe that adding this extra example would add significant value to the main text — after all, the point of Figure 11 (previously Figure 9) is to demonstrate the conservative properties of the superdroplet motion are correct in a divergence free flow. The following extract contains this new analysis (Supplement Section 2; P2 L17-23):

  "Figure 2 shows the same plot as Figure 11 of the main manuscript except that we run the simulation for 2 hours and include the terminal velocity of the tracers according to Rogers et al. (1993). We also model condensation/evaporation with a 1s timestep to make the tracers grow enough to have an appreciable fall speed but not so large that they fall out of the domain. The thermodynamics are time-invariant throughout the simulations, with pressure and temperature following hydrostatic profiles with surface values of 1013.15hPa and 289K respectively, and the domain is fully supersaturated throughout (supersaturation = 1.0).".

3. Here we have sub-divided point #3 regarding Cleo's parametrisations to aid readability:

   - The beginning and end parts of point #3 said:
     - *Please clarify and justify the choice of parameterizations, including when CLEO is designed to switch between different parameterizations, and when a single parameterization has been fixed. For instance, condensation and ventilation appear to have a single fixed option, whereas coalescence leaves room for multiple implementations.*
     - *It would further be useful to understand where CLEO's chosen parameterizations differ from those adopted by other production-ready SDMs (LibCloud, SCALE, PySDM). Differences between PySDM and CLEO in collision implementation are well-documented, but understanding any differences in CLEO's default choice for other dynamics such as the collision kernel and sedimentation velocity would be helpful to the modeling community.*

     Also in response to three comments in RC#2 regarding Cleo's assumptions/limitations, we have included three new sub-sections, one for each of the sections describing Cleo's numerical methods for condensation/evaporation, droplet collisions, and droplet motion. In these sub-sections we've tried to include an explicit and comprehensive list of all the model limitations and assumptions we think warm-cloud microphysics modellers may view as significant, as well as justify why we use fixed/variable parametrisations, and, where appropriate, mention how Cleo's parametrisations compare to other SDMs. Additionally, in the conclusion we have added a paragraph (P20 L523-530) summarising the most important points from these sub-sections. In the following, we quote the additions in these sub-sections and in the conclusion which relate specifically to the two parts of point #3 listed above.

     The default is actually that Cleo holds parametrisations fixed, since (P20 503-505) "Cleo is a concise and fully-functioning novel implementation of SDM for warm-cloud microphysics. Our primary motivation for creating Cleo is to provide a computationally efficient, "standard" SDM, which we could use in large-enough-domain LES to decipher the interactions between warm-cloud microphysics and shallow mesoscale cloud organisation."; and therefore (P20 L528-531) "All Cleo's limitations and assumptions make it poorly applicable to highly detailed microphysics studies requiring more complex processes, but make it well-designed, as intended, as a simplistic, ordinary representation of warm-cloud microphysics according to SDM.". In the sub-section on "Condensation/Evaporation: Limitations" we now justify why some parametrisations are made flexible: (P6 L152-156) " Whilst Cleo provides multiple options for the choice of terminal velocity (Section 4) and collision outcomes and probability calculations, as well as the ability to easily incorporate new ones (Section 3), Cleo's condensation/evaporation algorithm is comparatively inflexible. This reflects the greater consensus in modelling condensation/evaporation in contrast to the large uncertainty in droplet collisions and the multitude of acceptable droplet terminal velocity parametrisations in the literature. However [...]".

     Regarding the differences between Cleo's chosen parameterisations and those of other existing SDMs:

     - In the Sub-Section "Condensation/Evaporation: Limitations", following on from the above extract we now state (P6 L156-166) "However, this means that unlike for example PySDM, Cleo has a

fixed definition of its ventilation coefficient, arguably the most uncertain aspect of the ODE for condensation/evaporation. It also currently does not support a parametrised version of CCN activation, for example Twomey CCN activation as done by Grabowski et al. (2018), or a different numerical method for solving the condensation/evaporation ODE (e.g. as in Dziekan et al., 2019). Again in comparison to PySDM, Cleo does not provide between different parametrisations of thermodynamic formulae, for example for calculating supersaturation, conductivity or diffusion factors, or specific/latent heat capacities. (Nonetheless, if it were desired, a user could always change these fixed formulae or algorithms directly in the source code.)
  Additionally we do not account for any local perturbation in the environment around each superdroplet; they all experience the same grid-box values of thermodynamics and winds (Section 5), and Cleo's current representation of aerosols is crude, following Shima et al. (2009), unlike some SDMs which have since incorporated multiple aerosols and aerosol chemistry (Jaruga and Pawlowska, 2018; Bartman et al., 2022b).".

  - In the Sub-Section "Collisions: Limitations", we now state (P13 L311-314) ". Our decision to follow the original collision algorithm of Shima et al. (2009) means we are restricted to using a linear sampling for superdroplet pairs (although this is unlikely to be an issue; Dziekan and Pawlowska, 2017), and we do not implement any sub-time-stepping algorithm during collisions — for example as done in the latest version of PySDM (de Jong et al., 2023b)."
  - In the Sub-Section "Droplet Motion: Limitations", we now state (P14 L345- P15 L355) "As already mentioned, Cleo uses a simple second-order method for droplet motion and does not yet incorporate any stochastic element to represent sub-grid scale turbulent effects on droplet motion. For simulations with increasingly high resolution and low superdroplet multiplicities, it may become more desirable to use a higher order method or one with adaptive sub-time-stepping, as done by some other SDMs (e.g. Naumann and Seifert, 2015; de Jong et al., 2023b), because in such simulations the superdroplet positions more precisely correspond to the real droplets' positions. Especially in coarse resolution simulations, neglecting sub-grid scale turbulence in droplet motion is idealistic, and so in a later version of Cleo our method should be extended to include a stochastic component, e.g. following Dziekan et al. (2019). During superdroplet motion we also make two more minor assumptions, first, we neglect any momentum exchange between the air and droplets (unlike Shima et al., 2020), and second, we assume each particle reaches their terminal velocity instantaneously, since this is unlikely to be noticeable in LES (Naumann and Seifert, 2015). Our method for superdroplet motion is therefore most similar to the other SDMs described by Arabas et al. (2015); Jaruga and Pawlowska (2018), Chandrakar et al. (2021), and Matsushima et al. (2023).".
  - In the conclusion we have added that (P20 L523-530) "Cleo's numerical methods have some limitations in comparison to some existing SDMs. Regarding condensation/evaporation, perhaps the most significant is that Cleo's method for condensation/evaporation currently does not account for multiple aerosol types. Regarding collisions, we have not incorporated any sub-time-stepping algorithm. Whilst Cleo's simple predictor-corrector method for droplet motion is less expensive than higher-order methods and still preserves the flow-field, is non-diffusive, and has increasing accuracy with increasing grid resolution, it does not yet incorporate turbulent effects and is less precise than a higher order method. All Cleo's limitations and assumptions make it poorly applicable to highly detailed microphysics studies requiring more complex processes, but make it well-designed, as intended, as a simplistic, ordinary representation of warm-cloud microphysics according to SDM.".

  Also, to clarify that there is no default for the flexible parametrisations in Cleo, we now state in P8 L193-194: "There is no default calculation for the collision probability in Cleo, the user must always specify it during compilation.", and likewise in P14 L333-334: "Also analogously to the collision probability, there is also no default calculation for the terminal velocity in Cleo, the user must always specify which one to use during compilation.".

- The middle part of point #3 said:
  - *Please explain and justify the mixing of various implementation components, such as describing the probabilities of coalescence/breakup/rebound from Testik while utilizing the coalescence efficiency of Straub 2010.*

  We agree we did not explain this well enough and have revised the manuscript accordingly. Testik specifies three regimes but does not specify a coalescence efficiency. However, we need a coalescence efficiency for the regimes where coalescence and either breakup or rebound can occur. We therefore chose to use the coalescence efficiency from Straub et al. (2010). In revisiting this we also realised a more fitting reference is to Testik (2009) rather than Testik et al. (2011). We've therefore corrected this as well as rephrased the text to make clearer our reasoning for combining Testik (2009) and Straub et al. (2010). When we introduce this option for determining the outcome of the extended collision algorithm (P8 L204-206), we now state "Our algorithm can therefore easily interchange various formulations from the literature, at the time of writing from Low and List (1982a), **Testik (2009) combined with Straub et al. (2010)**, and Szakáll and Urbich (2018).", and similarly we now always state "Testik (2009) combined with Straub et al. (2010)" everywhere else we mention this choice of parametrisation combination. To explain why we combined the two parametrisations we now say (P8 L209-210) "Then, according to Testik (2009), there are three possible regimes depending on the magnitude of the collision kinetic energy, T_{E}, relative to the surface tension energy of the smaller and larger droplet, S_S and S_L respectively." followed up by the introduction of Straub et al. (2010) in P9 L220-223: "In regimes 1 and 2 a coalescence efficiency, $E_{\mathrm{c}} \in [0, 1]$, is required to give the probability two droplets coalesce, $P_{\mathrm{c},jk}$, given that they collided, i.e. $P_{\mathrm{c},jk} = E_{\mathrm{c}} P_{jk}$. Here we use the parametrisation for E_{c} as argued for by Straub et al. (2010), namely that [...]".

4. *Please describe some implementation details that are relevant to coupling with LES or another dynamical model:*

   We have added a Section (Section 5; P15 L355- P16 L374), which provides details of the coupling with LES / other models (strictly speaking Cleo's coupling to a "host dynamical driver" as we have defined it in Section 5 in accordance with its definition in Bayley et al. (2025)). Specifically in answer to the points this Referee made we have included the following:

   - *What are the parameterized source and sink terms to Eulerian tracers including latent heating, moisture/buoyancy sources, or any momentum feedbacks during transport?*

     (P16 L367-368) "Cleo then enacts SDM microphysics and superdroplet motion, which potentially alters the temperature, water vapour [mass mixing ratio] and liquid water [mass mixing ratio] in each grid-box but not the pressure nor winds (e.g. condensation increases temperature and converts water vapour to liquid).", and then later in the paragraph (P16 L372-374) "Any potential advection of thermodynamic fields, momentum feedbacks, radiation or turbulence schemes are the responsibility of the particular host dynamical driver which Cleo is coupled to.".

   - *At a minimum, it would be helpful to describe the prognostic variable sets that are natively compatible with the current implementation of CLEO, and which would require interpolation and/or additional coupling intricacies. For instance, many large scale models operate in pressure coordinates rather than z; others have different choices of moisture and thermodynamic prognostic variables. In my experience, converting between these native prognostic variables and those required by the SGS scheme can result in qualitative differences simply due to small differences in a choice of thermodynamic constants, so it is helpful to understand the native variable set that is most compatible with CLEO.*

     (P15 L359-364) "Cleo's grid can be any kind of Arakawa C-grid and is composed of grid-boxes which define the volumes of the domain and the necessary Eulerian thermodynamic and wind fields so that the Lagrangian superdroplets can enact microphysics and move around. These are: the temperature, pressure, water vapour mass mixing ratio, and liquid water mass mixing ratio defined at the centre of each of Cleo's grid-boxes, as well as the wind velocity defined on the faces of each grid-box. (The meridional wind component is defined at the centre of the faces perpendicular to north-south, and analogously for the zonal and vertical wind components.)".

     Concerning your comment on pressure/z coordinates: Exactly for the reason you mention we defined Cleo such that there is no pre-defined pressure or z coordinate. In fact, Cleo is more general than this because there is no pre-defined coordinate system at all — Cleo only specifies that there are up to 3 dimensions, i.e. directions, (which may or may not be orthogonal) as defined when choosing a particular Arakawa C-grid implementation. This is explained in detail in Bayley et al. (2025) Section 2.1.

Minor Comments

- *The CLEO acronym is never defined*

  The acronym no longer applies as we have renamed "CLEO" to "Cleo", also explained in the response to RC#2: "We think the acronym we gave Cleo in the original (companion) manuscript was not only unhelpful but also restrictive for Cleo's future development. We've therefore reverted to using Cleo as a proper noun, as it was originally intended to be, in tribute to the real-life "super-women", Cleopatra and the anonymous mathematician whose pseudonym was Cleo.". This origin of Cleo's name is explained in the revised version of the companion to this manuscript (egusphere-2025-4398).

- *L92-93: "Droplets with a radius... same terminal velocity" needs a reference or proof.*

  We made this statement by assessing the parametrisations Cleo has for droplet terminal velocities. We've therefore added an explicit reference to Figure 6 in P5 L119: "(see also Figure 6)"and in Figure 6's caption we now also state: "In all the formulations, droplets with a radius larger than approximately 3 *mm* have the same terminal velocity.".

- *L129: "references in Morrison et al. 2020" is not an appropriate citation; select the appropriate supporting references and cite them directly*

  We're corrected this by citing (P7 L172-173) "Hu and Srivastava, 1995; McFarquhar, 2004; Stevens and Seifert, 2008; Straub et al., 2010; Morrison and Milbrandt, 2011; Planche et al., 2019".

- *Please clarify the rationale for terminating a simulation rather than removing an empty superdroplet (L207)*

  In P11 L257-260 we've added our explanation: ."We do this because terminating the simulation rather than removing the superdroplet simplifies our algorithms and we do not expect this case to occur in the simulations of mesoscale cloud organisation which Cleo is intended for (which would only contain superdroplets with very large multiplicities). In a future version of Cleo we may amend our algorithms to make Cleo also suitable for simulations of superdroplets with $\mathcal{O}(1)$ multiplicities.".

- *I could not find the citation for Bayley 2025b; Bayley 2025a needs an updated URL with the relevant DOI*

  We've updated both these references now the pre-prints are available. (This is also explained in our response to RC#2 "we've updated the references to the pre-prints for Bayley et al. 2025 (this manuscript's companion manuscript; egusphere-2025-4398) and for Niebaum et al. 2025 (egusphere-2025-5551). We've also updated the references to the final accepted versions of de Jong et al. (2023a) and Matsushima et al. (2023).").

- *L228: the result of collision is only deterministic in certain cases in CLEO, but is still probabilistic in that it uses the random number (Monte Carlo) in other cases*

  We realise we didn't explain this carefully enough. What we meant to say was that the step to determine whether or not a collision occurs is indeed probabilistic (Monte Carlo), but what happens afterwards can use the same random number again or use the superdroplet attributes to determine the outcome, rather than draw a new random number. In those cases we meant "deterministic" in that the collision outcome (rebound, coalescence or breakup) is already determined once it has been decided that a collision occurs. To clarify this on P12 L279-281 we've added "[...] although the step to determine if a collision does or does not occur is always probabilistic (Section 3), whether a collision then results in rebound, coalescence, or breakup in our algorithm does not have to depend on drawing a new random number, whereas in de Jong et al. (2023a) an additional probabilistic step is always used.".

- *Can you clarify for a non-computer scientist what a C++20 concept is?*

  A C++20 concept can be thought of as a list of constraints a code-object must satisfy in order for some C++ code to compile. In the context of this paper it means a class/structure has a function with a specific signature. We have reworded the sentences that mention C++20 concepts to briefly outline what they are (at least for our purposes), or at the very least make the sentences readable for non-computer science readers. We now say on P8 L189-191 "The function to calculate P_{jk} in Cleo can therefore be changed before compilation as long as it still obeys a set of technical constraints on the function's signature (defined by a C++20 concept (ISO, 2020); see also Bayley et al., 2025)."and on P14 L330-332 "In Cleo the terminal velocity is defined, analogously to the collision probability (Section 3), by a set of technical constraints on the signature of the function which calculates it (using a C++20 concept (ISO, 2020); see also Bayley et al., 2025), meaning that different formulations can be easily interchanged.".

- *L358 and elsewhere: clarify that adaptive sub-time-stepping is only implemented in condensation, not in coalescence (whereas PySDM v2 includes a coalescence adaptive time step)*

  We have corrected P6 L146 to "adaptive sub-time-stepping"instead of "adaptive timestepping", and in the "Collisions: Limitations" sub-section we now explicitly state (P13 L313-314) "we do not implement any sub-time-stepping algorithm during collisions — for example as done in the latest version of PySDM (de Jong et al., 2023b).". In the "Droplet Motion: Limitations" sub-section we also added (P15 L346-347) "it may become more desirable to use a higher order method or one with adaptive sub-timestepping, as done by some other SDMs (e.g. Naumann and Seifert, 2015; de Jong et al., 2023b), [...]".

  To avoid any confusion, we also now mention that Cleo does enable adaptive time-stepping *between* microphysical processes with different time-steps: (P21 L532-534) "Simulations can be up to 3-D and can include any combination of the primary microphysical processes behind warm-rain, each with their own independent time-step (and with adaptive time-stepping between different processes, as explained in Bayley et al., 2025).".

- *The lightest yellow in Figure 7 is difficult to see and does not print well.*

  We have replaced the yellow in Figure 7 with a dark red.

- *A table of notation would be helpful for reference*

  We have now added this in the two tables in Appendix B (Table B1 and Table B2).

Technical Corrections

- *L32: "effect" → affect*: corrected.
- *L110: "upto"*: corrected, also elsewhere.
- *L147: "we prescribe for the collision probability" – a leftover/hanging phrase?*: Yes we apologise for the oversight and have removed it.
- *L312: "was" → were*: corrected.