Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **547**, 1–27 (2026) https://doi.org/10.1093/mnras/stag368
Advance Access publication 2026 February 23

# On the consistency of jet feedback modelling across different astrophysics hydrodynamical codes

N. Maragkakis,[1,2★] M. A. Bourne,[3,4,5] C. Power,[1,2] F. Huško,[6] A. Ludlow[1] and S. Shabala[7]

[1] *International Centre for Radio Astronomy Research, The University of Western Australia, 35 Stirling Highway, Crawley, Western Australia 6009, Australia*
[2] *ARC Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D)*
[3] *Centre for Astrophysics Research, Department of Physics, Astronomy and Mathematics, University of Hertfordshire, College Lane, Hatfield, AL10 9AB, UK*
[4] *Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK*
[5] *Kavli Institute for Cosmology (KICC), University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK*
[6] *Leiden Observatory, Leiden University, PO Box 9513, NL-2300 RA Leiden, the Netherlands*
[7] *School of Natural Sciences, University of Tasmania, Private Bag 37, Hobart, Tasmania 7001, Australia*

Accepted 2026 February 16. Received 2026 January 19; in original form 2025 October 22

## ABSTRACT

Active Galactic Nuclei (AGNs) feedback is essential in cosmological simulations of galaxy formation, yet its implementation has to rely on subgrid models due to limited resolution. We present a novel subgrid jet-launching method for galaxy formation simulations and implement it in three hydrodynamical codes: the smoothed particle hydrodynamics (SPH) code SWIFT, the moving-mesh code AREPO, and the Eulerian grid code PLUTO. To isolate the impact of hydrodynamical solvers on jet evolution, we compare idealized jets and their remnants in uniform and stratified media across resolutions and jet parameters. In uniform media, all jets drive bow shocks, inflate hot lobes, exhibit backflows, and evolve self-similarly. For the parameters explored, SWIFT lobes are shorter, wider, and hotter; AREPO lobes are longer, thinner, and cooler; while PLUTO lobes display complex flows with intermediate characteristics. In stratified media, jets deviate from self-similar evolution, inflating longer and thinner lobes due to lower external ram pressure. After switch-off, SWIFT jets evolve into smooth cylindrical bubbles, AREPO jets produce long filamentary remnants, and PLUTO jets yield intermediate-length remnants with varying degrees of mixing. Despite such differences, all jets and remnants have a similar impact on the ambient medium. We conclude that variations in lobe properties between codes emerge even for identical subgrid prescriptions, since the coupling of jet feedback to resolvable scales and the effective resolution depend on the hydrodynamical method. In structure formation simulations, these solver differences are likely subdominant to uncertainties in subgrid modelling and calibration, while averaging over galaxy populations may lessen their impact.

**Key words:** methods: numerical – galaxies: formation – galaxies:jets.

## 1 INTRODUCTION

Numerical simulations are a powerful tool for the theoretical astrophysicist, allowing them to study highly complex and non-linear problems. This is particularly true in modelling galaxy formation, where feedback – the coupling of mass, momentum, and energy injected by stars and black holes into their ambient medium – can drive behaviour that is highly sensitive to physical conditions on a range of spatial and temporal scales (M. Vogelsberger et al. 2020). Cosmological hydrodynamical simulations and semi-analytic models have revealed that feedback plays a key role in shaping the properties of the galaxy population, such as the galaxy luminosity function (e.g. S. Cole et al. 2000; A. J. Benson et al. 2003; R. G. Bower et al. 2006; D. J. Croton et al. 2006) and (equivalently) the galaxy stellar mass function (GSMF) and the stellar-to-halo mass relation (e.g. J. Schaye et al. 2014; Y. Dubois et al. 2016; I. G. McCarthy et al. 2016; A. Pillepich et al. 2017; N. A. Henden et al. 2018; R. Davé et al. 2019). Additionally, feedback contributes to establishing scaling relations such as those between black hole mass and stellar bulge mass (e.g. C. M. Booth & J. Schaye 2010; J. Schaye et al. 2014; D. Sijacki et al. 2015; M. Volonteri et al. 2016), galaxy size and mass (e.g. J. Schaye et al. 2014; M. Furlong et al. 2016; S. Genel et al. 2017; Arjona-Gálvez, Elena et al. 2025), and H I mass and halo mass (e.g. G. Chauhan et al. 2020). Feedback from active galactic nuclei (AGNs) is now generally accepted as a key mechanism for ensuring that star formation in massive galaxies is quenched (and remains so), and that galaxies attain colours and morphologies consistent with observations (e.g. Y. Dubois et al. 2013, 2016 ; D. Sijacki et al. 2015; M. Donnari et al. 2020; L. Byrne et al. 2024; P. H. Goubert et al. 2024).

AGN feedback is driven by the accretion of matter onto the central supermassive black holes of galaxies, and its effects are

★ E-mail: nick.maragkakis@research.uwa.edu.au

evident in galaxy groups and clusters (e.g. A. Fabian 2012; D. Eckert et al. 2021). It acts in a variety of ways: driving accretion-disc winds (A. R. King & K. A. Pounds 2003; K. A. Pounds et al. 2003; A. King & K. Pounds 2015; T. Costa, R. Pakmor & V. Springel 2020); exerting direct radiation pressure (W. Ishibashi & A. C. Fabian 2015; R. Bieri et al. 2017; T. Costa et al. 2018); and/or launching fast bipolar jets (P. A. G. Scheuer 1974; M. J. Hardcastle & J. H. Croston 2020; M. A. Bourne & H.-Y. K. Yang 2023; D. Mukherjee 2025), which heat the ambient medium (S. Y. Sazonov et al. 2005; F. Arrigoni Battaia et al. 2018) and eject massive outflows (D. S. N. Rupke & S. Veilleux 2011; J. S. Spilker et al. 2025). AGN jets can contribute to the quenching of star formation in galaxies and the regulation of heating and cooling in galaxy groups and clusters (e.g. B. McNamara & P. Nulsen 2007; A. Fabian 2012; J. Hlavacek-Larrondo, Y. Li & E. Churazov 2022; M. A. Bourne & H.-Y. K. Yang 2023). Evidence for such AGN jet feedback is provided by X-ray cavities that are filled with radio-emitting plasma, observed in galaxy clusters, such as Perseus (H. Böehringer et al. 1993; A. C. Fabian et al. 2000, 2003, 2006; Hitomi Collaboration 2016; Weeren, R. J. et al. 2024) and Virgo (W. Forman et al. 2005, 2007).

The theoretical basis for AGN feedback as a key regulator of galaxy growth was originally outlined by J. Silk & M. J. Rees (1998). AGN feedback in galaxy formation modelling was then subsequently included in semi-analytic models as radio-mode feedback. This injected energy into the hot gaseous atmospheres surrounding massive galaxies in high-mass haloes, offsetting gas cooling and preventing them from becoming overly massive and luminous (R. G. Bower et al. 2006; D. J. Croton et al. 2006). It differed from hydrodynamical simulations, which modelled AGN feedback as quasar-mode feedback via thermal energy injections (e.g. T. Di Matteo, V. Springel & L. Hernquist 2005; T. Di Matteo et al. 2008). While such models are still in use, many modern hydrodynamical galaxy formation simulations adopt a two-mode AGN feedback scheme, which combines the quasar and radio modes – at high (low) black hole accretion rates, the radiatively efficient (inefficient) quasar (radio) mode is active (D. Sijacki et al. 2007, 2015; Y. Dubois et al. 2012; R. Weinberger et al. 2016; R. Davé et al. 2019).

Despite this progress, implementing AGN feedback in hydrodynamical simulations remains challenging because the relevant physical processes span a large spatial range, from sub-parsec to megaparsec scales (see e.g. M. Gaspari, F. Tombesi & M. Cappi 2020; M. S. S. L. Oei et al. 2024). The limited resolution of cosmological galaxy formation simulations means that AGN feedback, by necessity, is included as a subgrid model whose primary aim is to capture its impact on scales resolvable in the computational domain. This demands simplifying prescriptions for how feedback is deposited into the gas surrounding black holes, and it is commonplace to calibrate models to reproduce present-day ($z = 0$) large-scale observables such as the GSMF (J. Schaye et al. 2014; A. Pillepich et al. 2017; R. Davé et al. 2019), the black hole mass–stellar bulge mass relation (J. Schaye et al. 2014; Y. Dubois et al. 2016; R. Davé et al. 2019), and, more recently, the baryon content of groups and clusters (I. G. McCarthy et al. 2016; N. A. Henden et al. 2018; R. Kugel et al. 2023; L. Bigwood et al. 2025). Moreover, assumptions made in galaxy formation models (e.g. the choice of cooling function or star formation model) as well as the underlying hydrodynamical solver, will affect the implementation and physics of feedback (M. A. Bourne, K. Zubovas & S. Nayakshin 2015). Together, these factors contribute to substantial uncertainties in the first-principles modelling of AGN feedback in cosmological hydrodynamical simulations.

At the same time, variations in feedback models and parameter choices can lead to differences in the resulting galaxy populations. While global galaxy statistics across large-volume cosmological simulations – such as EAGLE (J. Schaye et al. 2014), Horizon-AGN (Y. Dubois et al. 2016), IllustrisTNG (A. Pillepich et al. 2017), and SIMBA (R. Davé et al. 2019) – are in broad agreement in terms of stellar mass functions and star formation rate densities, discrepancies emerge at the level of individual systems. For example, galaxy morphologies (S. Genel et al. 2019), cold gas fractions (R. Davé et al. 2020), black hole growth histories (M. Habouzit et al. 2021), and gas inflows and outflows around central galaxies in halos (R. J. Wright et al. 2024) can differ significantly between simulations. Moreover, additional differences arise in the predicted population statistics of quasars and AGN jets, as quantified by the quasar and AGN radio luminosity functions (N. Fanidakis et al. 2010; M. Raouf et al. 2017; N. Thomas et al. 2021).

This frames the twofold challenge that galaxy formation simulations face. First, AGN feedback cannot be modelled from first principles on galactic scales and requires subgrid models. Secondly, the effects of these models are highly sensitive to numerical choices and parameters, making interpretation and comparison of results difficult. In this context, comparison projects involving different hydrodynamical codes and different models of cooling, star formation, and feedback are a powerful numerical tool to help us address these challenges.

Accurate modelling of AGN jets requires a careful treatment of strong shocks and fluid instabilities, both of which are sensitive to the underlying hydrodynamical method. In general, the choice of hydrodynamical solver can impact the outcome in several key aspects of astrophysical problems. Because of this, there have been a number of studies exploring the differences between SPH, moving-mesh/meshless, and grid-based approaches in hydrodynamical simulations (e.g. C. S. Frenk et al. 1999; G. M. Voit, S. T. Kay & G. L. Bryan 2005; O. Agertz et al. 2007; J. W. Wadsley, G. Veeravalli & H. M. P. Couchman 2008; D. Sijacki et al. 2012; C. Power, J. I. Read & A. Hobbs 2014; P. F. Hopkins 2015; F. Sembolini et al. 2016). Some studies have examined standard hydrodynamical tests (such as a dense gas cloud impinging on a diffuse wind, the so-called blob test) and showed that grid codes (O. Agertz et al. 2007) and moving-mesh/meshless codes (D. Sijacki et al. 2012; P. F. Hopkins 2015) perform better than traditional SPH codes. However, recent improvements in the SPH methods (e.g. S. Rosswog & D. Price 2007; J. I. Read & T. Hayfield 2012; T. R. Saitoh & J. Makino 2013; A. M. Beck et al. 2015; J. W. Wadsley, B. W. Keller & T. R. Quinn 2017; D. J. Price et al. 2018; J. Borrow et al. 2021a; T. Sandnes et al. 2025) make them increasingly competitive in their performance against standard benchmarks (e.g. J. Braspenning et al. 2023).

Traditionally, grid codes have handled interacting fluids, mixing, and instabilities better (O. Agertz et al. 2007) while being able to sufficiently resolve high-density contrasts, making them the preferred computational method for high-resolution AGN jet feedback simulations (M. A. Bourne & H.-Y. K. Yang 2023). Building upon this, moving-mesh and meshless codes have the potential to improve upon the fixed-mesh approaches and, when combined with refinement schemes, have proven effective at resolving high-resolution jets in idealized (M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017; K. Ehlert et al. 2022) as well as more realistic cluster environments (M. A. Bourne, D. Sijacki & E. Puchwein 2019; M. A. Bourne & D. Sijacki 2021; K.-Y. Su et al. 2021). More recently, an SPH code has been used to model high-resolution idealized jets in F. Huško & C. G. Lacey (2023a), and jets and remnants in F. Huško & C. G. Lacey (2023b), highlighting the ease and efficiency of particle-based methods.

In this paper, we examine the AGN jets produced by three independent, state-of-the-art, astrophysical codes:

(i) SWIFT (M. Schaller et al. 2024), which represents the Lagrangian or smoothed particle hydrodynamics approach,
(ii) AREPO (V. Springel 2010), which represents the moving-mesh approach, and
(iii) PLUTO (A. Mignone et al. 2007, 2012), which represents the Eulerian or adaptive mesh-refinement approach.

To do this, we employ a novel feedback model that is designed for resolutions achievable in modern cosmological and galaxy formation simulations.

We note that pure Eulerian codes are still the most widely used approach in AGN jet simulations and are well suited to capturing the complex structure of jet lobes, including jet recollimation, backflows, and turbulence (e.g. P. M. Yates-Jones, S. S. Shabala & M. G. H. Krause 2021; P. M. Yates-Jones et al. 2023). In contrast, moving-mesh codes require specialized refinement schemes to properly resolve the low-density lobes (e.g. M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017), while SPH codes are expected to be less effective for jet simulations without improvements in mixing and particle-splitting techniques. None the less, it is useful to quantify any differences between codes and test different parameters and resolutions to assess their consistency. This is especially relevant for cosmological hydrodynamical simulations where different numerical approaches may offer unique advantages and where AGN jet modelling is only one aspect of more complicated systems.

In Section 2, we briefly review the key features of the three codes we consider: SWIFT, AREPO, and PLUTO. In Section 3, we model our idealized AGN jets using the aforementioned codes with our novel launching module. We present our results in Section 4, including uniform-medium runs, stratified-medium runs and remnant runs. Finally, we discuss our results in Section 5, and summarise our conclusions in Section 6.

## 2 ASTROPHYSICAL HYDRODYNAMICAL CODES

We use the public versions of three astrophysical codes, considering only pure hydrodynamics coupled to a minimal jet model (see Section 3.2).

### 2.1 SWIFT

SWIFT (M. Schaller et al. 2024) is a Lagrangian, SPH code that solves the equations of motion for the fluid using particle-carried and kernel-smoothed quantities. We adopt the default SPHENIX (J. Borrow et al. 2021b) scheme that incorporates artificial viscosity to capture shocks, as well as artificial conductivity to promote mixing between different fluid layers. Both are supplemented with limiters that adapt their strength accordingly, to avoid excessive viscosity in shear flows and unwanted energy dissipation in feedback injections. Additionally, we make use of the particle-splitting module in SWIFT to keep all SPH particle masses roughly the same throughout the simulations.

### 2.2 AREPO

AREPO (V. Springel 2010; R. Pakmor et al. 2016; R. Weinberger, V. Springel & R. Pakmor 2020) is a quasi-Lagrangian, moving-mesh code that solves hydrodynamics on an unstructured Voronoi mesh using a second-order, finite-volume discretisation with an exact Riemann solver. The mesh can move with the flow, and in its standard operating state, a roughly constant mass resolution is maintained via a cell refinement and de-refinement scheme. Previous approaches implemented within AREPO to study AGN-driven jet evolution (e.g. M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017; R. Y. Talbot, M. A. Bourne & D. Sijacki 2021) have employed additional super-Lagrangian refinement criteria to enhance resolution within the jets and lobes. However, we use the standard quasi-Lagrangian approach (i.e. fixed-mass resolution) to ensure our model remains applicable to a broad range of galaxy formation simulations.

### 2.3 PLUTO

PLUTO (A. Mignone et al. 2007, 2012) is an Eulerian, grid-based simulation code developed for high-Mach-number astrophysical fluid flows. The fluid is evolved on a static three-dimensional Cartesian grid by solving the conservation laws using the HLLC (Harten–Lax–van Leer–Contact) approximate Riemann solver with linear reconstruction, and a second-order Runge–Kutta time-integration scheme. To increase the simulation robustness, we use the shock-flattening feature to switch to the HLL solver and the MINMOD limiter in the presence of strong shocks. The code supports several different physics modules, including special relativistic hydrodynamics and magnetohydrodynamics. However, we only use non-relativistic hydrodynamics in the absence of magnetic fields to ensure broad applicability.

## 3 JET FEEDBACK THEORY AND MODELLING

We focus on simulating idealized hydrodynamical jets in simplified environments to facilitate robust code comparisons. To maintain control over the set-up, we do not couple jet injection to accretion onto a supermassive black hole but, instead, introduce jets with fixed powers and durations, using parameters set by hand. These include jet power, injection velocity, opening angle, and ambient-medium properties, chosen to produce jets that resemble Fanaroff–Riley type II (FR-II) sources and inflate lobes of hot gas. This idealized approach enables a clean and controlled comparison between codes, avoiding complications arising from differences in accretion physics. While the jets may exhibit features of real AGN sources, our set-up is deliberately simplified to isolate the effects of jet propagation and lobe inflation. This allows us to determine their impact on the thermodynamic profiles of stratified media, and to explore implications for cosmological simulations.

We adopt an adiabatic equation of state and neglect gas cooling, self-gravity, and magnetic fields. Additionally, our jets are injected with sub-relativistic speeds ($< 0.5\,c$). Although launching jets at relativistic speeds and including special relativistic effects can influence jet stability and even lobe properties (S. S. Komissarov & S. A. E. G. Falle 1998; W. English, M. J. Hardcastle & M. G. H. Krause 2016; M. Perucho et al. 2017; P. M. Yates-Jones et al. 2023), classical jets capture important aspects of the evolution, especially the self-similar scaling of lobes (see e.g. R. J. Turner & S. S. Shabala 2023). This is relevant for cosmological galaxy formation simulations that typically do not implement special relativistic effects.

### 3.1 Self-similar theory

Our simulations consist of jets propagating in ambient media with power-law density profiles of the form

$$\rho = \rho_0 \left(\frac{r}{r_c}\right)^{-\alpha}, \tag{1}$$

where $\rho_0$, $r_c$, and $\alpha$ are constants. We consider two cases with $\alpha < 2$ to study jets, lobes, and remnants. We include runs in a uniform medium with $\alpha = 0$, and in a stratified medium with $\alpha \sim 1$. We compare our jets with each other and with an analytic self-similar solution that is derived by making a number of simplifying assumptions. In the self-similar regime, the jet length $L_{\rm jet}$ depends only on the jet power, $P_{\rm j}$, and the ambient-medium density, as the product $\rho_0 r_c^\alpha$. Using dimensional analysis, we can derive its evolution with time as (S. A. E. G. Falle 1991)

$$L_{\rm jet} \sim l \equiv \left(\frac{P_{\rm j}}{\rho_0 r_c^\alpha}\right)^{1/(5-\alpha)} t^{3/(5-\alpha)}. \tag{2}$$

For a uniform medium, $\alpha = 0$, and hence

$$L_{\rm jet} \sim l = \left(\frac{P_{\rm j}}{\rho_0}\right)^{1/5} t^{3/5}. \tag{3}$$

The exact formula, valid for cylindrical jets, can be found by including a dimensionless constant of order unity, $C$, in the above formula as (C. R. Kaiser & P. Alexander 1997; S. S. Komissarov & S. A. E. G. Falle 1998; C. R. Kaiser & P. N. Best 2007)

$$L_{\rm jet} = C \times l = C \left(\frac{P_{\rm j}}{\rho_0}\right)^{1/5} t^{3/5}, \tag{4}$$

where

$$C = \left\{\frac{A^4}{18\pi} \frac{(\gamma+1)(\gamma-1)(5-\alpha)^3}{9[\gamma + (\gamma-1)A^2/2] - 4 - \alpha}\right\}^{1/(5-\alpha)}. \tag{5}$$

Here, $\gamma = 5/3$ is the ideal-gas adiabatic index and $A$ is the lobe aspect ratio (i.e. the ratio of the jet-lobe length to the jet-lobe width). In the self-similar solution, the lobe aspect ratio remains constant throughout the jet's evolution. We estimate $A \sim 1/\theta$, where $\theta$ is the half-opening angle of the jet (S. S. Komissarov & S. A. E. G. Falle 1998). Then the width of the jet can be expressed as

$$W_{\rm jet} = \frac{L_{\rm jet}}{A}. \tag{6}$$

While this analytic solution ignores several key features of AGN jets, like the breakout phase, backflows, instabilities, and non-thermal pressure (e.g. from cosmic rays), it is nevertheless a useful, approximate guide for the large-scale evolution of the jet lobes. There are other analytic and simulation based models of jet propagation that might better approximate specific aspects of the evolution (e.g. the early phase) (M. C. Begelman & D. F. Cioffi 1989; M. J. Hardcastle 2018; K.-Y. Su et al. 2021; R. J. Turner et al. 2023), but comparing these to our results is beyond the scope of this paper.

### 3.2 Jet injection model

In designing a model for jet injection, it is necessary to account for the inherent differences between grid and SPH codes. There is a plethora of grid-based models in the literature that launch jets by injecting mass, momentum, and energy (e.g. H. Y. K. Yang & C. S. Reynolds 2016; M. A. Bourne & D. Sijacki 2017; D. Martizzi et al. 2018) or by setting a hydrodynamic state within the injection region (e.g. R. Weinberger et al. 2017, 2023; P. M. Yates-Jones et al. 2023). On the other hand, SPH jets can be realized via injection or kicking of particles (F. Huško et al. 2022; F. Huško & C. G. Lacey 2023a, b). To ensure a clean comparison between different codes, we developed a hybrid virtual-particle model that is largely code agnostic. Virtual particles act as carriers of mass, momentum, and energy, but are hydrodynamically decoupled; they are launched from the origin and deposit the mass, momentum, and energy into the ambient medium once certain criteria are satisfied.

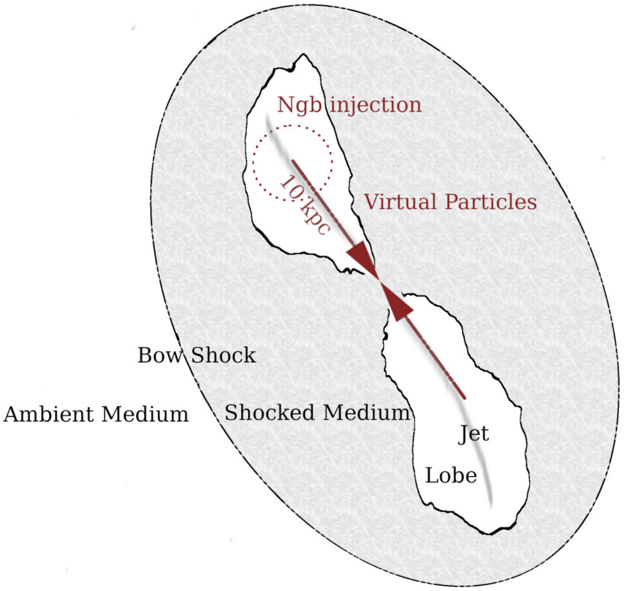

**Figure 1.** Sketch of our virtual-particle model for a single injection event. Two virtual particles are launched in opposite directions, inside two cones, and travel for 10 kpc without interacting with the medium, before they inject mass, momentum, and energy into their gas neighbours. Denoted are the different components of the resulting jets and lobes they inflate.

We simulate a single jet episode with a series of discrete injection events. A single injection event consists of launching two virtual particles from the centre of our computational domain along opposite directions (with the first chosen at random), inside two coaxial cones of half-opening angle $\theta = 15^\circ$ to produce two jets and ensure conservation of linear momentum. The virtual particles travel with constant velocity, without interacting with the gaseous medium, until they reach a pre-defined radius, $r_0 = 10$ kpc. They then deposit mass, momentum, and energy into the 10 nearest gas elements, in a mass-weighted fashion, and are removed from the computational domain (see Fig. 1).

A pair of virtual particles is launched whenever

$$P_{\rm j}\Delta t \geq 2 \times \frac{1}{2} m_{\rm j} v_{\rm j}^2, \tag{7}$$

where $P_{\rm j}$ is the jet power, $\Delta t$ is the time elapsed since the last injection event, and $m_{\rm j}$ and $v_{\rm j}$ are the virtual-particle (jet) mass and velocity, respectively. The term $1/2\, m_{\rm j} v_{\rm j}^2$ represents the energy carried by each particle, noting that the factor of two in equation (7) accounts for two particles being generated and kicked in opposite directions. This energy is not added into the system until the virtual particles have propagated to the prescribed distance $r_0$.

Upon reaching this radius, each particle's 10 nearest gas elements receive mass and energy as

$$\Delta m = m_{\mathrm{j}}, \tag{8}$$

and

$$\Delta E = \frac{1}{2} m_{\mathrm{j}} v_{\mathrm{j}}^2, \tag{9}$$

so that the total mass and total energy injected (from both particles) are 2 × $\Delta m$ and 2 × $\Delta E$. Each neighbouring gas element $i$ receives a fraction of this feedback, updating its mass and energy as

$$m_i = m_{i0} + \frac{m_{i0}}{m_{\mathrm{ngb}}} \times m_{\mathrm{j}}, \tag{10}$$

$$E_i = E_{i0} + \frac{m_{i0}}{m_{\mathrm{ngb}}} \times \frac{1}{2} m_{\mathrm{j}} v_{\mathrm{j}}^2, \tag{11}$$

where $m_{i0}$ and $E_{i0}$ are its initial mass and energy, and $m_{\mathrm{ngb}}$ represents the total mass of the neighbours in a loop.

For this energy to be interpreted as kinetic, we add momentum accordingly to achieve the proper final state. Following M. A. Bourne & D. Sijacki (2017), the momentum of each gas element is updated as

$$\boldsymbol{p_i} = \boldsymbol{p_{i0}} + \left(\sqrt{2 m_i K_i} - |\boldsymbol{p_{i0}}|\right) \frac{\boldsymbol{v_{\mathrm{j}}}}{v_{\mathrm{j}}}. \tag{12}$$

where $\boldsymbol{p_{i0}}$ is its initial momentum, $K_i$ is the target kinetic energy ($E_i$ minus any initial thermal energy), and the resulting momentum kick acts in the direction of travel of the virtual particle. As a small caveat, this formula does not strictly conserve the injected energy due to possible momentum cancellation. In this case, an additional thermal component is added to account for the difference (see M. A. Bourne & D. Sijacki 2017, for details).

The final velocities of the gas elements will then depend on their initial masses and kinetic energies, as well as on the parameter $v_{\mathrm{j}}$, and will thus vary between the codes. Nonetheless, for simplicity, we will refer to the parameter $v_{\mathrm{j}}$ as the (jet) injection velocity.

To faithfully follow the jet's propagation in the surrounding medium, a proper numerical integration scheme must also be chosen with care. For AGN feedback to be accurately captured in simulations, time-stepping must be fine enough to resolve its effects within the feedback region; for instance, this has been shown to be an important source of error in simulations of supernova feedback (e.g. T. R. Saitoh & J. Makino 2009; F. Durier & C. Dalla Vecchia 2012). To control hydrodynamical time steps, we use the standard Courant–Friedrichs–Lewy (CFL) condition, which relies on the sound speed and size of a resolution element to set a time-step limit. However, the CFL condition does not account for the extra injection of energy and supersonic propagation of jets when resolution elements first receive feedback.

Thus, for codes that employ hierarchical, local time-stepping (SWIFT and AREPO), we enforce small and fixed time steps for all resolution elements that reside in the central regions ($r <$ 40 kpc) of our domain and will be directly or indirectly affected by feedback injections. Moreover, we include the built-in time-step limiter in SWIFT, of the kind advocated by T. R. Saitoh & J. Makino (2009), to employ a neighbour loop that wakes up inactive particles if their time steps are too large compared to their neighbours (M. Schaller et al. 2024). Similarly, we include the non-local tree-based time-step limiter in AREPO, which aims to estimate the arrival time of waves from other locations within the simulation domain (V. Springel 2010). On the other hand, for the global time-stepping code PLUTO we adopt a fixed time step for all cells, and set it small enough to accommodate jet feedback. In this set-up, all codes have a fixed, uniform, and small enough time step for the resolution elements that are affected by feedback, as well as their neighbours. We note that the purpose of this study is the careful modelling and comparison of jets in idealized environments, and we reserve performance and scalability practicalities, including more specialized time-stepping schemes for realistic astrophysical or cosmological environments, to future works.

Finally, an important consideration when developing an AGN feedback model is refinement. While moving grid (AREPO) and fixed grid (PLUTO) codes can use refinement and adaptive grids to achieve higher resolution in specific areas of interest, SPH codes (SWIFT) do not employ these techniques. This is relevant for high-resolution jet simulations and isolated jet studies, which are traditionally carried out with grid codes that employ a variety of refinement criteria or adaptive grid techniques to better resolve the propagation of jets and their interaction with the environment (e.g. A. Cattaneo & R. Teyssier 2007; Y. Li & G. L. Bryan 2014; H. Y. K. Yang & C. S. Reynolds 2016; M. A. Bourne & D. Sijacki 2017; M. A. Bourne & H.-Y. K. Yang 2023). However, in cosmological structure formation simulations, excessive refinement inside the jet region can become computationally expensive and/or present unintended negative consequences for other aspects of the simulation. For our results to be relevant for such cosmological simulations, we choose not to include any additional specialized refinement techniques in AREPO, nor an adaptive grid in PLUTO. Nonetheless, to ensure particles in SWIFT retain similar masses after mass injections (which is important for SPH calculations), we include particle-splitting and split particles that reach twice their initial mass. Similarly, cells in AREPO are refined/de-refined to ensure that they have a consistent mass within a factor of two. These choices facilitate a fair comparison of codes under comparable effective resolutions, without introducing bias from extra refinement methods.

### 3.3 Initial conditions and jet parameters

For all simulations, we use a periodic box with a volume of 300 × 300 × 600 kpc$^3$ with the jet propagating along the long $z$-axis. In our uniform-medium runs, resolution elements are arranged on a regular Cartesian grid with initial density $\rho_0 = 2.5 \times 10^{-26}\,\mathrm{g\,cm^{-3}}$ and initial temperature $T = 10^7$ K; this environment is representative of the central regions of galaxy clusters. We choose these parameters so as to produce jets that achieve self-similar behaviour (with various degrees of deviation from the analytic solution). In addition to the uniform medium, in Sections 4.5 and 4.6, we perform simulations of jets and remnants in a stratified medium following an initially isothermal beta profile

$$\rho = \rho_0 \left[1 + \left(\frac{r}{r_c}\right)^2\right]^{-3\beta/2} \tag{13}$$

with $\rho_0 = 10^{-25}\,\mathrm{g\,cm^{-3}}$, $\beta = 0.38$ and $r_{\mathrm{c}} = 10$ kpc. At large distances from the origin $\rho \sim r^{-3\beta} = r^{-1.14}$, so that the analytic self-similar solution remains valid. The beta-density profile is motivated by observations of X-ray emission from galaxy groups and clusters and is frequently used in simulations of AGN jets (e.g. J. C. Vernaleo & C. S. Reynolds 2006; W. English et al. 2016; P. M. Yates-Jones et al. 2021). In this case, we include an external

**Table 1.** Summary of all jet simulations performed using SWIFT, AREPO, and PLUTO..

| | Medium | Box volume (kpc$^3$) | Resolution elements | Injection velocity (km s$^{-1}$) | Neighbour loop scheme |
|---|---|---|---|---|---|
| Fiducial | Uniform | $300^2 \times 600$ | $10^8$ | $4 \times 10^4$ | Neighbour number |
| Resolution study | Uniform | $300^2 \times 600$ | $10^7, 5 \times 10^7, 10^8$ | $4 \times 10^4$ | Neighbour number |
| Velocity study | Uniform | $300^2 \times 600$ | $10^8$ | $2 \times 10^4, 4 \times 10^4, 8 \times 10^4$ | Neighbour number |
| Neighbour mass study | Uniform | $300^2 \times 600$ | $10^8$ | $2 \times 10^4, 4 \times 10^4, 8 \times 10^4$ | Neighbour mass |
| Stratified medium jets | Stratified | $300^2 \times 600$ | $2.9 \times 10^7$ | $4 \times 10^4$ | Neighbour number |
| Remnants | Stratified | $900^2 \times 1200$ | $1.8 \times 10^7$ | $4 \times 10^4$ | Neighbour number |

gravitational field with acceleration

$$\mathbf{g} = -3\beta u \frac{\gamma - 1}{r^2 + r_c^2}\mathbf{r} \tag{14}$$

where $u = \frac{1}{\gamma-1}\frac{k_b T}{\mu m_p}$ is the initial internal energy, $T = 10^7$ K is the initial temperature, constants have their usual meaning, and we set $\mu = 0.6$. This potential is computed analytically so as to ensure that the initial configuration is in hydrostatic equilibrium.

Throughout this work, we set the jet power to $P_j = 5 \times 10^{45}$ erg s$^{-1}$, typical of moderate to high-power radio-galaxy jets. In our standard, uniform-medium runs (Section 4.1), we set the virtual-particle velocity to $v_j = 4 \times 10^4$ km/s to ensure the jets enter the self-similar regime. We set the virtual-particle mass to $m_j = 4 \times 10^4$ M$_\odot$ to fix the number of injection events. This virtual-particle mass will be safely smaller than the masses of individual gas elements in SWIFT and AREPO to help keep the mass resolution relatively constant. The jets remain active for 98 Myr, resulting in approximately 12 000 injection events.

In Section 4.2, we compare different resolution runs in the uniform medium, in Section 4.3, we vary the injection velocity $v_j$, and in Section 4.4, we compare with simulations that inject into a fixed neighbour mass. In the stratified-medium runs (Sections 4.5, 4.6), we halve the virtual-particle mass to $m_j = 2 \times 10^4$ M$_\odot$, keep the same velocity and shorten the jet duration to 44 Myr to simulate a similar number of injection events and produce similar-length jets. In the uniform medium we use resolutions that vary from $N = 10^7$ to $N = 10^8$ resolution elements whereas in the stratified medium we instead employ $N = 2.9 \times 10^7$ and $1.8 \times 10^7$ resolution elements. We summarise the complete set of our runs in Table 1.

## 4 RESULTS

### 4.1 Comparison of jets across codes

To compare the large-scale evolution and morphology of jets across different hydrodynamical solvers, we now examine their properties in our fiducial simulation set.

*Visual impression:* In the top panel of Fig. 2, we show slices[1] in the $y = 0$ plane, for the jets simulated with the three codes after 98 Myr in the uniform medium. These simulations represent our highest resolution runs, consisting of $10^8$ initial resolution elements that have an initial mass of $M_{\rm gas} = 2 \times 10^5$ M$_\odot$. Each slice is split into four parts depicting gas temperature (top left), density (top right), pressure (bottom left), and velocity magnitude (bottom right).

All of the jets drive a bow shock into the ambient medium and inflate lobes of hot gas. We can readily distinguish the different components of the jets (as labelled in Fig. 1): a sheath of high-velocity jet material; high-temperature, low-density lobes consisting of shocked jet material; the shocked ambient medium behind the bow shock; and the yet unshocked ambient medium. All jets and lobes appear coherent, with any instabilities not significantly disrupting their structure. Inspecting the pressure slices, we infer that the jet lobes and the shocked external medium behind the bow shock achieve rough pressure equilibrium (especially in AREPO and PLUTO), even though the jet heads remain overpressured. In the velocity slices, the jet material in the central sheath reaches velocities very close to the jet injection velocity (i.e. the virtual-particle velocity), whereas the shocked material populating the jet lobes exhibits lower velocities.

There are some distinctions between jets simulated with different codes. SWIFT jets inflate short and hot lobes that are wide at the base. In contrast, AREPO jets produce longer, cooler, and thinner lobes. Finally, PLUTO lobes are of intermediate length and width and relatively hot. Both AREPO and PLUTO lobes show Kelvin–Helmholtz (KH) instabilities between the lobe and the shocked ambient medium that are not resolved at this resolution in SWIFT.

Each code solves the fluid equations with a specific discretisation method that may influence the dynamics and propagation of jets. In the bottom panel of Fig. 2, we zoom in on the white box region in the top panel in the temperature slice and depict the underlying structure of each of the codes. SWIFT lobes are resolved with particles with smoothing kernels while AREPO uses a voronoi mesh and PLUTO a static grid. We observe how SWIFT and AREPO achieve variable spatial resolution based on the local density, with lower spatial resolution inside the lobes and higher spatial resolution at the interface of the lobe and ambient medium. In contrast, PLUTO has the same spatial resolution everywhere. These different set-ups will also influence the development of fluid instabilities, with higher lobe resolution generally allowing for more complex flows and smaller-scale KH instabilities (R. Weinberger et al. 2023; F. Huško & C. G. Lacey 2023a).

A key feature of our simulations is the presence of backflows. Backflows consist of jet material that, after being shocked at the jet head, flows backwards along the jet axis, toward the injection site. In the radial velocity maps in Fig. 3, we can see that all jets exhibit high-velocity backflows around the central, outward-flowing sheath. SWIFT jets show the least coherent backflows, while AREPO jets display strong, streamlined backflows, and PLUTO jets produce fast but irregular backflows.

[1] The slices for the SWIFT jets are taken using the SWIFTSIMIO python module (J. Borrow & A. Borrisov 2020) which accounts for SPH smoothing. The slices for the AREPO and PLUTO jets are generated directly from local cell properties within the slice plane.

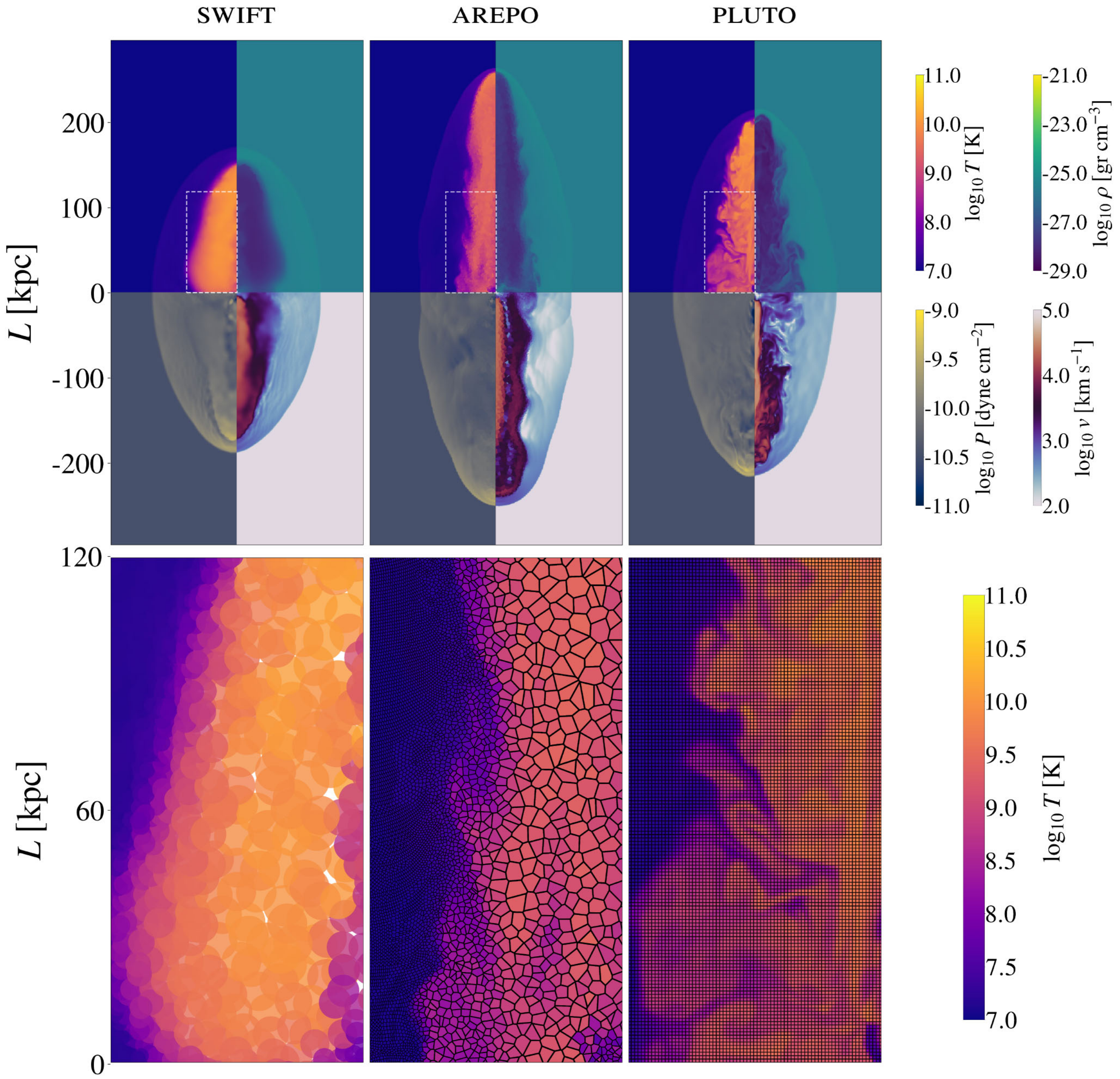

**Figure 2.** Comparison of SWIFT, AREPO and PLUTO jets. **Top:** Overview slices in the $y = 0$ plane in the uniform medium after 98 Myr at our highest resolution ($N = 10^8$, $M_{\rm gas} = 2 \times 10^5\,{\rm M}_\odot$) simulations. Mapped quantities are temperature (top left), density (top right), pressure (bottom left), and velocity magnitude (bottom right). All jets drive a bow shock in the ambient medium, inflate lobes, and develop backflows. SWIFT jets inflate shorter, wider, and hotter lobes, AREPO jets produce longer, thinner, and cooler lobes, and PLUTO jets yield lobes that are of intermediate length and width and relatively hot. **Bottom:** Zoomed-in (white box in the top panel) temperature map displaying the underlying discretisation method for each code. We can distinguish differences between codes in the spatial resolution of the lobes and the medium, as well as in the development of Kelvin–Helmholtz instabilities.

The observed backflows can also influence the development of KH instabilities, whose growth rate scales with the velocity shear.

In the case of PLUTO, the central sheath of jet material reaches the highest velocities, up to the jet injection velocity ($v_{\rm j} = 40,000\,{\rm km\,s^{-1}}$). This is a direct consequence of our jet injection model; our injection scheme distributes kinetic energy and mass to a fixed number of gas elements (see Section 3.2). In SWIFT and AREPO, these elements maintain roughly constant mass throughout the simulation, due to particle-splitting or mesh refinement, but in PLUTO, cells can reach arbitrarily low densities, and hence masses, especially in the centre of the domain as the jet evacuates material. Consequently, injecting the same kinetic energy and mass into lower-mass cells leads to higher effective jet velocities in PLUTO, compared to the other codes. A different, less code-dependent behaviour is observed when, instead of injecting the jet into a fixed number of neighbouring gas elements, we inject it into a fixed mass of gas elements. This results in more consistent effective jet velocities, with the AREPO and PLUTO jets appearing more similar (see Section 4.4).

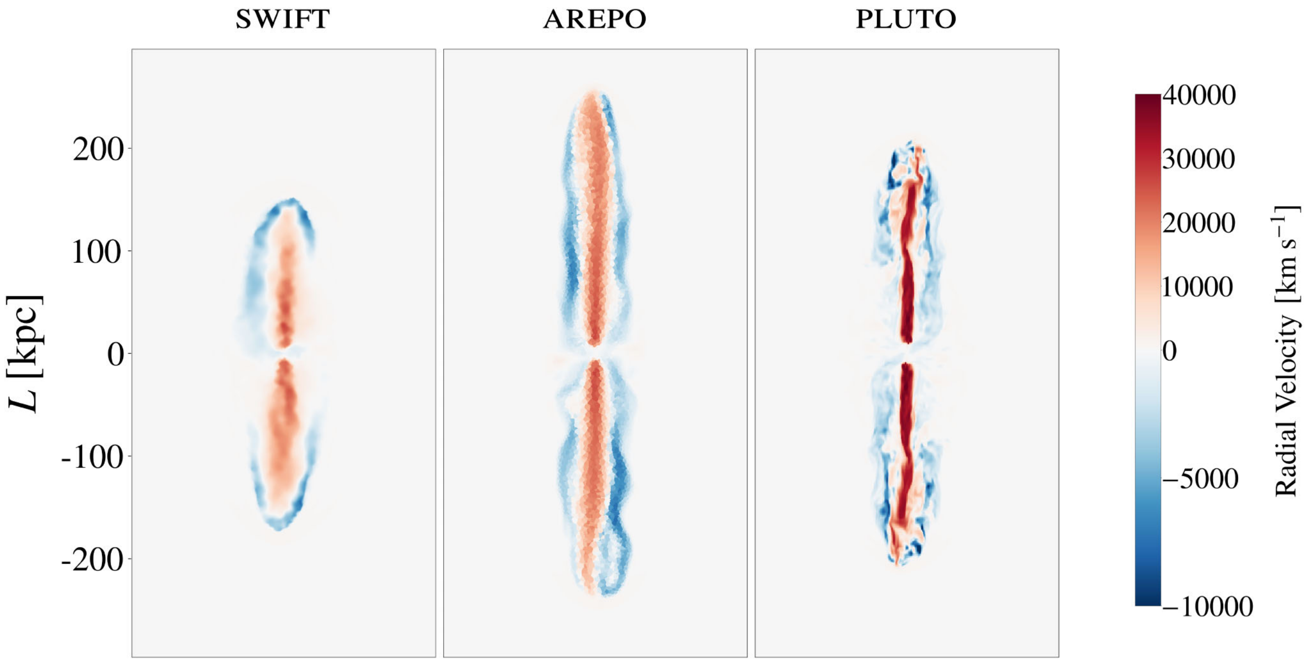

**Figure 3.** Radial velocity slices of jets simulated with the three codes in the uniform medium after 98 Myr. All jets display a central sheath of high-velocity material moving outwards along the jet axis, surrounded by backflows. SWIFT jets show the least coherent backflows, AREPO jets display stronger and more streamlined backflows, and PLUTO jets present fast but irregular backflows.

The higher-velocity flows in PLUTO, in combination with the higher resolution in the lobes, may be causing the jets to disrupt before reaching the termination shock, as the jet head is more susceptible to KH instabilities (P. M. Yates, S. S. Shabala & M. G. H. Krause 2018). These instabilities could develop between the jet and the shocked lobe material due to the high-velocity shears at the jet head. In real jets, however, magnetic fields may act to stabilize the fluid against KH instabilities (e.g. P. E. Hardee 2007).

While the final state of the jets and their lobes provides useful information, examining their time evolution across the three codes can give us some insight into the dynamics driving their expansion. To this end, in Fig. 4 we display temperature (top) and density (bottom) slices for the three jets in the uniform medium, for progressively later times in each row. SWIFT lobes appear to retain their shape from quite early in the simulation while progressively growing in size, indicating self-similar expansion. In contrast, the lobes simulated with AREPO and PLUTO become noticeably elongated in the first two rows (for the first ∼ 50 Myr) but mostly retain their shape during their later evolution. We also note, to varying degrees, the progressive fusing of the top and bottom lobes at their base in all the codes. This could be attributed to the development of backflows but also to the growth of some modes of KH instabilities at the base of the AREPO and PLUTO lobes. Finally, we observe the development of multiple recollimation shocks in the PLUTO jet by 50 Myr. These shocks seem to disrupt by the end of the simulation, as the flow becomes more susceptible to instabilities. *Jet lobe evolution:* To quantify the differences between jets simulated with different codes, we calculate the lengths and widths of the jet lobes.

(i) **Jet lobe** material is identified as elements (particles/cells) that have temperatures $T > 10^8$ K or velocity magnitudes $v > 0.5 \times 10^4$ km/s.

(ii) **Jet length** is defined as the average distance to the furthest 1 per cent of lobe resolution elements from the origin along the jet axis ($z$).

(iii) **Jet width** is defined as the average perpendicular distance of the 1 per cent of elements that lie furthest in the transverse direction $\left(\sqrt{x^2 + y^2}\right)$. Note that this yields the maximum width of the lobe.

Using the 1 per cent of the number of elements in each lobe, rather than the 1 per cent of mass or volume, is less physically motivated but conceptually simpler. In any case, we do not find any significant differences between the methods. The final values are the averages between the two lobes for each simulation. We checked that our estimates are in rough agreement with those obtained using a jet tracer in AREPO and PLUTO, where lobe material is defined by tracer values $tr > 0.01$.

In Fig. 5, we plot the lobe lengths (solid lines) and widths (dotted lines) of the jets simulated with different codes, in different colours, and include the self-similar analytic solution, equations (4)–(6). Black lines represent the lobes simulated with SWIFT, blue lines the AREPO lobes, and red lines the PLUTO lobes and the dashed grey line the analytic solution. All of the codes produce lobes that expand self-similarly by the end of the simulations. Specifically, for lobe length, SWIFT underpredicts, AREPO overpredicts, and PLUTO agrees with the normalized analytic solution. At the same time, all codes overpredict the lobe width compared to the normalized solution for most of the duration of the jet episode.

Although agreement with the slope of the self-similar solution is a robust indicator of self-similar behaviour, we note that agreement with the normalized solution, in practice, depends on the particular combination of jet parameters (F. Huško & C. G. Lacey 2023a), especially jet injection velocity (see Section 4.3).

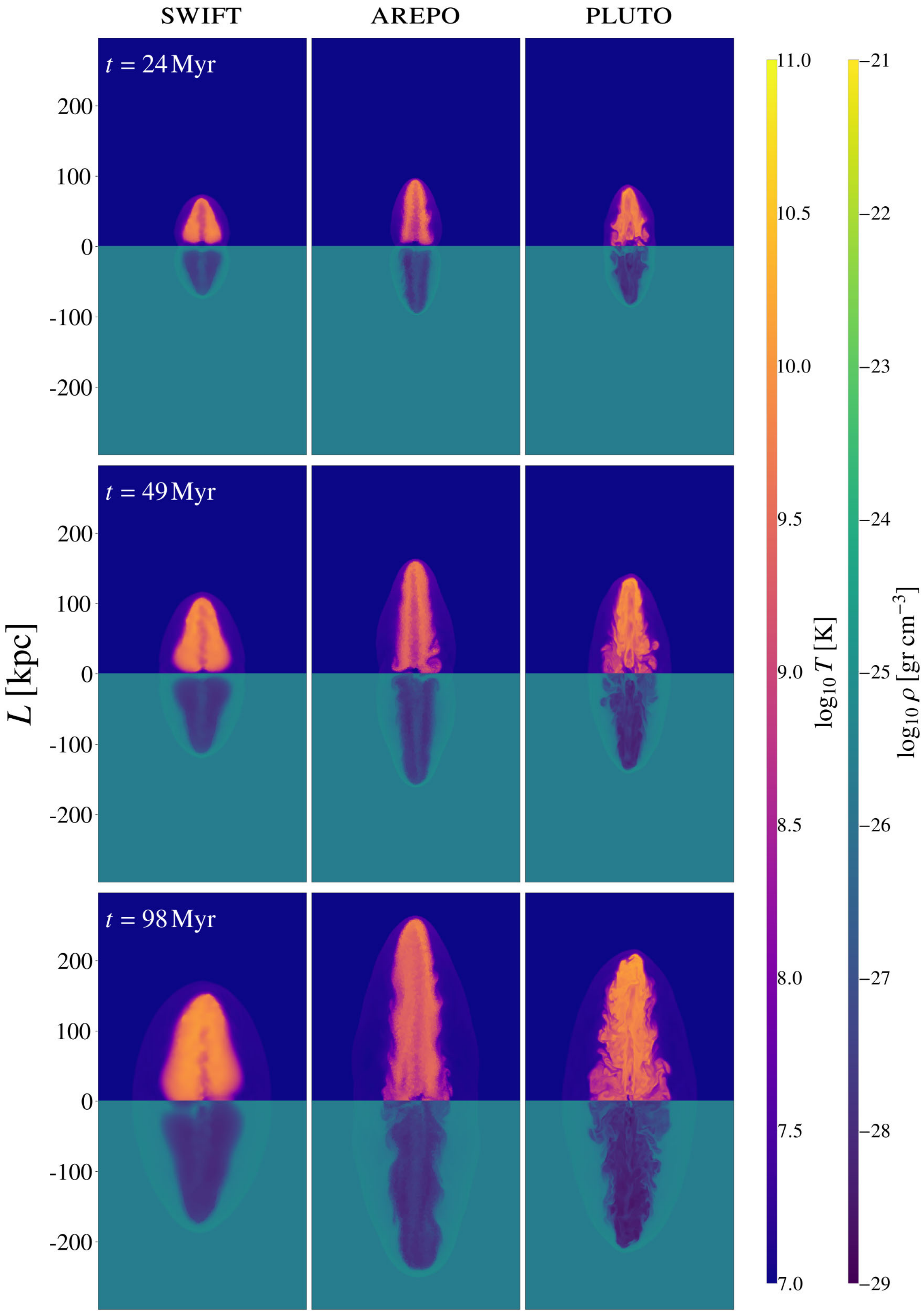

**Figure 4.** Temperature (top) – density (bottom) slices of the time evolution of the jets simulated with the three codes in the uniform medium. Different rows showcase the different jets at progressively later times. The SWIFT lobes retain their shape and evolve self-similarly from early on, while AREPO and PLUTO lobes are becoming more elongated, for at least the first 50 Myr. The top and bottom lobes progressively fuse in all codes due to backflows and the growth of KH instabilities.

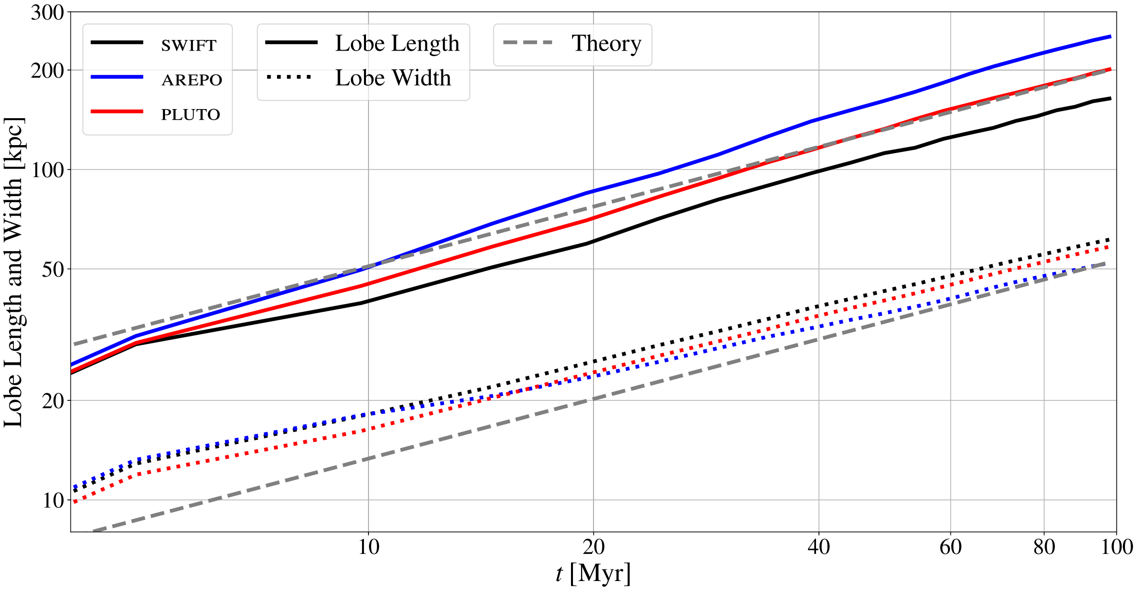

**Figure 5.** Plots of lobe length and width with respect to time for jets simulated with the three codes in the uniform medium. Black lines represent the lobes simulated with SWIFT, blue lines the AREPO lobes, and red lines the PLUTO lobes. Solid lines represent estimates of length, and dotted lines represent estimates of width. The dashed grey lines represent the analytic, self-similar solution. All jets produce lobes that expand self-similarly by the end of the simulations. SWIFT underpredicts, AREPO overpredicts, and PLUTO agrees with the normalized analytic solution for lobe length.

To more clearly demonstrate the self-similar evolution of the jets, we estimate how their lobe aspect ratio (the lobe length divided by the width) changes as the simulation progresses. In the first row of Fig. 6, we plot the lobe aspect ratio with respect to time for the jets simulated with the three codes (solid coloured lines) and the analytic solution (dashed grey line). SWIFT lobes appear to expand self-similarly from very early on since their aspect ratio does not change significantly throughout the simulation, in line with the analytic solution being just a constant. In contrast, AREPO and PLUTO jets showcase a more obvious, transitional, breakout phase (S. S. Komissarov & S. A. E. G. Falle 1998; M. J. Hardcastle & M. G. H. Krause 2013; F. Huško & C. G. Lacey 2023a) where the jets inflate long and thin lobes that increase their aspect ratios – this breakout phase is not captured by the self-similar solution. Eventually, the lobes start exhibiting self-similar behaviour after around 40–60 Myr.

In the second row of Fig. 6, we plot the lobe expansion speeds of the different jets and include the analytic prediction. This speed is calculated by evaluating the change in lobe length between consecutive snapshots and dividing by the time difference. Naturally, this bulk lobe speed will always be lower than the jet velocity as the jet material shocks against the ambient medium at the jet head. The three codes show good agreement with each other and the normalized analytic result, with SWIFT lobes being marginally slower, PLUTO lobes somewhat faster, and AREPO lobes the fastest. All of the lobes decelerate over the duration of the simulation and end up having a similar speed of $\sim 1500\,\mathrm{km\,s^{-1}}$ by the end.

The last two rows of Fig. 6 show the time evolution of the lobe volume (top) and lobe mass (bottom) for jets simulated with the three codes and the self-similar prediction for lobe volume. All simulation codes show monotonic growth in both volume and mass, consistent with ongoing jet activity. PLUTO produces lobes with the largest volumes, while SWIFT and AREPO yield smaller lobe volumes that are in better agreement with the analytic solution. For the last simulation snapshot, PLUTO overpredicts lobe volume by 20 per cent compared to the analytic solution, while SWIFT and AREPO agree to within 1 per cent and 4 per cent, respectively. In terms of lobe mass, we also show the mass injected by the jet (solid silver line), which represents the minimum possible lobe mass across all codes. SWIFT consistently predicts lower values than the other codes, indicating less efficient mixing and entrainment with the ambient medium.

Besides the dimensions and appearance of the lobes, a critical quantity to consider in the context of AGN feedback and its impact on galaxy, group and cluster evolution is the lobe energetics. In Fig. 7, we plot the different energy components inside the lobes (dashed line) and in the ambient medium (solid line) with respect to time. We denote thermal energy in red, kinetic energy in blue, and total energy, which is the sum of the two, in black. We subtract any initial thermal energy to ensure that we capture only the injected energy. Overall, we find good agreement between the three codes; all of the jets transfer around 60 per cent of the injected energy to their ambient medium, with most of it being in thermal form. The energy in the lobes is predominantly thermal in all the codes, but with slightly different ratios; SWIFT and PLUTO lobes contain roughly four times more thermal energy than kinetic energy, whereas AREPO lobes have about twice as much. While almost all of the energy injection in our jets is in kinetic form, the high injection velocities efficiently thermalise the gas, inflating lobes that quickly become more thermal-energy-dominated (as opposed to kinetic-energy-dominated) and expand due to both their thermal pressure laterally and to their thrust in the jet direction. We expect these trends to still hold qualitatively if cooling is included, although there could be quantitative differences.

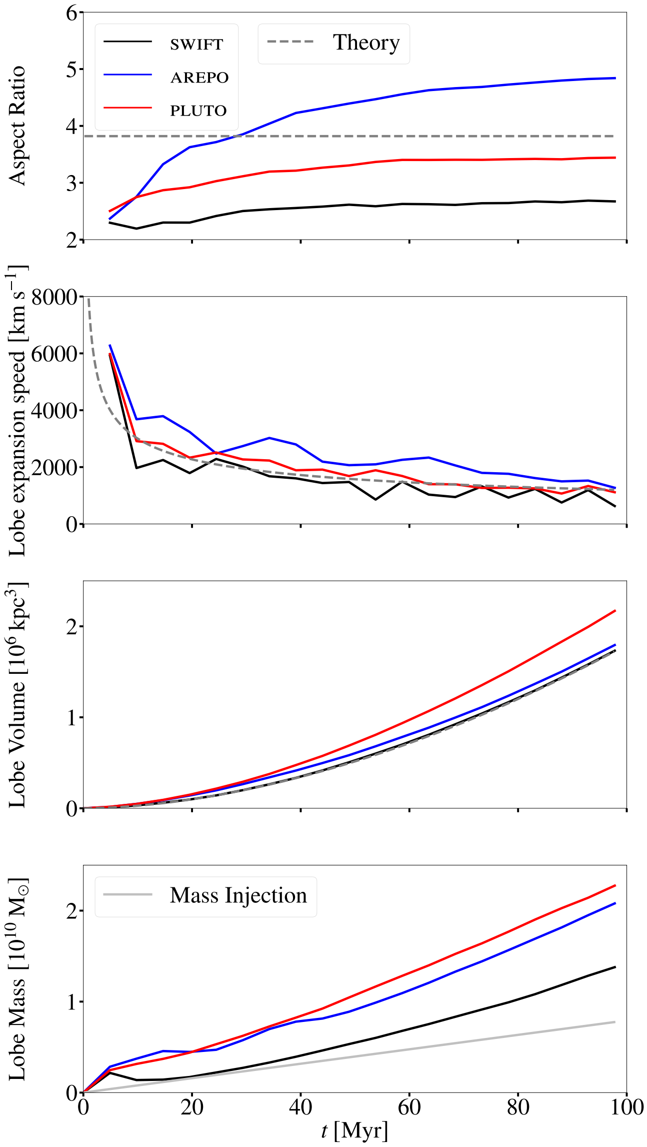

**Figure 6.** Lobe properties of jets produced by the three codes with respect to time. Solid coloured lines represent the three codes and dashed grey lines the analytic self-similar model. **First row:** Aspect ratio of jet lobes (lobe length divided by width) with respect to time. SWIFT lobes show minimal aspect ratio evolution with time, while AREPO and PLUTO lobes are increasing their aspect ratio for the first $\sim$ 40–60 Myr. **Second row:** Speeds of lobes with respect to time. All lobes are decelerating to around 1500 km s$^{-1}$ by the end of the simulation. **Third row:** Lobe volume as a function of time. SWIFT lobes match the analytic prediction closely. AREPO lobes reach slightly larger volumes, while PLUTO lobes consistently attain the largest volumes. **Fourth row:** Lobe mass as a function of time. The solid silver line represents the mass injected by the jet. AREPO and PLUTO lobes achieve comparable masses, whereas SWIFT lobes are slightly less massive, indicating less efficient mixing and entrainment.

### 4.2 Sensitivity to numerical resolution

When comparing different hydrodynamical codes, it is important to consider the inherent differences in how they discretise the fluid, as this impacts the effective mass and spatial resolutions achievable within a simulation.

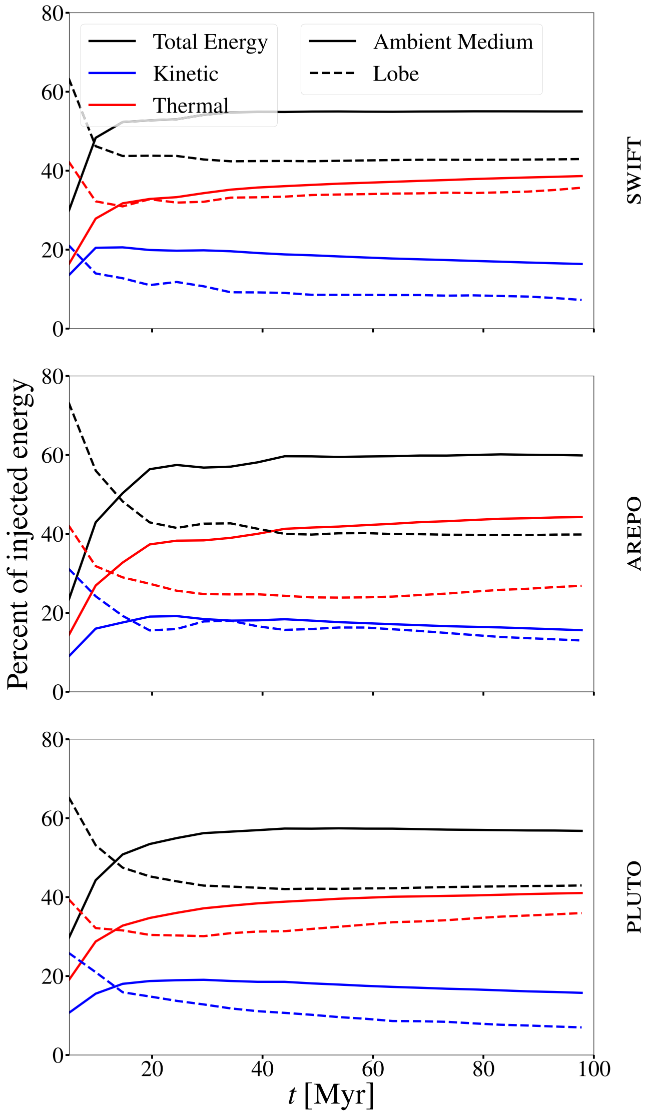

**Figure 7.** Evolution of the different energy components (thermal energy in red, kinetic energy in blue and total energy, as the sum of the two, in black) in the lobes (dashed line) or ambient medium (solid line) of the jets simulated with the three codes. All jets transfer around 60 per cent of their injected energy to the ambient medium, and all lobes contain more thermal energy than kinetic energy.

(i) **Mass resolution:** SWIFT and AREPO maintain roughly *constant mass resolution* (to within a factor of two) using particle-splitting and refinement, respectively. In contrast, PLUTO cells, which maintain a fixed volume, can achieve arbitrarily low (high) masses in low- (high-)density regions of a simulation, and hence present *adaptive mass resolution*. This is especially relevant for the inner parts of our simulation domain ($r$ < 10 kpc) where, due to the action of the jet, PLUTO cells achieve very low densities and hence masses. This, in turn, influences the injection and evolution of the jet (see Section 4.4).

(ii) **Spatial resolution:** SWIFT and AREPO feature *adaptive spatial resolution*. As an SPH code, SWIFT interpolates local fluid quantities via kernel smoothing over neighbouring particles, with spatial resolution set by the smoothing length; overlapping regions require independent smoothing lengths for a more conservative estimate (O. Agertz et al. 2007). AREPO, instead, employs a moving-mesh whose cells adapt to the fluid flow, with a mesh-regularization scheme that prevents excessive distortions and keeps them approximately round (V. Springel 2010); spatial resolution is then set by an effective cell radius $R_{\rm cell} = \sqrt[3]{3V_{\rm cell}/4\pi}$. In both codes, spatial resolution scales with local gas density: high-density regions are resolved more finely, while low-density regions are coarsely resolved. Consequently, in jet simulations, lobes are modelled at lower spatial resolution than the shocked ambient medium and bow shock (see bottom panel of Fig. 2). In contrast, in PLUTO, the use of a regular Cartesian grid with fixed cell size ensures *constant spatial resolution* in both lobes and ambient medium.

These differences imply that different hydrodynamical methods will be better suited to capturing different aspects of jet evolution, such as shock structure, lobe expansion, or instability growth, and should be considered carefully when interpreting simulation results.

To access differences in effective spatial resolution, we focus on the simulated jet lobes, whose low densities make them challenging to resolve with Lagrangian and quasi-Lagrangian codes. In the final snapshots of our fiducial simulations, within the lobes,

(i) SWIFT has smoothing lengths ranging from 1 kpc to 9 kpc (with mean 3 kpc)
(ii) AREPO has cell radii ranging from 0.5 kpc to 3.5 kpc (with mean 1.5 kpc)
(iii) PLUTO has cells with half-side 0.4 kpc.

Thus, PLUTO achieves the highest spatial resolution in the lobes, followed by AREPO, and finally SWIFT. Even in the unperturbed ambient medium, however:

(i) SWIFT has smoothing lengths of 1 kpc
(ii) AREPO has cell radii of 0.5 kpc
(iii) PLUTO has cells with half-side 0.4 kpc

This means that SWIFT starts with lower spatial resolution in the initial conditions of our simulations, due to SPH smoothing. A set-up of an equal number of resolution elements across codes, therefore inherently disadvantages SPH methods, and this must be considered when assessing their applicability. Nevertheless, such comparisons remain useful in the context of cosmological simulations, where the number of elements serves as a convenient proxy for the achieved mass and spatial resolution.

The specific resolution element count in the lobes varies between the codes: SWIFT lobes are resolved with 66 550 particles, AREPO lobes with 110 748 cells and PLUTO lobes with 4103 820 cells. As shown in Fig. 6, the total lobe mass and volume are very similar between the AREPO and PLUTO runs, while the difference in the number of lobe resolution elements across the codes reflects the interplay between mass and spatial resolution. In contrast, SWIFT lobes contain less mass, indicating reduced mixing and entrainment of ambient material – consistent with the apparent absence of KH instabilities along the SWIFT lobes. By construction, the lower levels of entrainment, and hence mass, in SWIFT lobes naturally result in fewer resolution elements within the lobes.

To illustrate the impact of resolution for each code, we present three sets of runs in the uniform medium, with different initial numbers of resolution elements (and initial gas element masses): $N = 10^7$ ($M_{\rm gas} = 2 \times 10^6\,{\rm M}_\odot$), $N = 5 \times 10^7$ ($M_{\rm gas} = 4 \times 10^5\,{\rm M}_\odot$), and $N = 10^8$ ($M_{\rm gas} = 2 \times 10^5\,{\rm M}_\odot$) resolution elements. We fix all other parameters, as well as the number of injection events, to be the same as in our fiducial runs. Each code begins with uniform spatial resolution and identical initial gas element masses, ensuring a consistent starting point. However, as the simulations evolve, differences in fluid discretisation lead to the code-specific behaviour we highlighted above. Specifically, SWIFT and AREPO progress from uniform spatial resolution to adaptive spatial resolution based on the local density. For PLUTO, the gas mass resolution can change significantly throughout the simulations depending on the local gas density.

Fig. 8 shows slices of temperature and density for SWIFT, AREPO and PLUTO jets in the uniform medium, at $t = 98$ Myr, with each row representing a different resolution. The simulations with $N = 10^8$, in the bottom row, represent our fiducial runs. Comparing the lobes at this final snapshot of the simulations:

SWIFT jets present only minor changes across different resolutions. Their lobes become slightly longer (+4 per cent) and slightly wider (+5 per cent) from lowest to highest resolution. This mild dependence on resolution agrees with F. Huško & C. G. Lacey (2023a), where SWIFT lobes showed noticeable shortening only at very low or high resolutions not probed here.

By contrast, AREPO jets show a stronger resolution dependence. Their lobes get notably longer (+14 per cent) from $N = 10^7$ to $N = 5 \times 10^7$, before becoming shorter again (−8 per cent) at $N = 10^8$. Their widths steadily increase (+15 per cent) from lowest to highest resolution. Increasing lobe length with resolution in AREPO jets has been noted in M. A. Bourne & D. Sijacki (2017) and R. Weinberger et al. (2017, 2023), while an eventual turnaround was observed in R. Weinberger et al. (2023).

Similarly, the lobes simulated with PLUTO become modestly longer (+4 per cent) from $N = 10^7$ to $N = 5 \times 10^7$, and then get significantly shorter (−18 per cent) at $N = 10^8$. Their width slightly increases (+5 per cent) at the highest resolution. This reduction of lobe length with resolution in PLUTO jets has been observed in P. M. Yates et al. (2018).

The variation of lobe length with respect to time for all our simulations (as summarized in Table 1) can be seen in Fig. 9. In the top row, we plot the lobe lengths of the jets at different resolutions. The coloured lines represent the three codes, with line style showing the specific resolution and the dashed grey line the analytic solution, equations (4) and (5). Across resolutions, SWIFT lobes are the shortest, consistently underpredicting the normalized analytic solution. AREPO and PLUTO lobes show better agreement with each other at lower resolutions while overpredicting the normalized solution. At the highest resolution, however, AREPO lobes remain quite long while PLUTO lobes shorten to match the normalized result.

It is worth noting that our use of a fixed-neighbour-number injection scheme leads to the following effect: higher resolutions have lower masses for individual gas elements and hence lower neighbour masses for injections of the same kinetic energy, which results in higher velocities. These higher velocities could then contribute to the decrease in lobe length observed in AREPO and PLUTO at the highest resolution ($N = 10^8$) as explained in Section 4.3 – although the effect seems to be subdominant at lower resolutions.

Besides the dimensions of the jets and their lobes, we can see how their morphologies change with increased resolution. One obvious effect is noted in the AREPO and PLUTO lobes, which more readily develop complex flows and KH instabilities

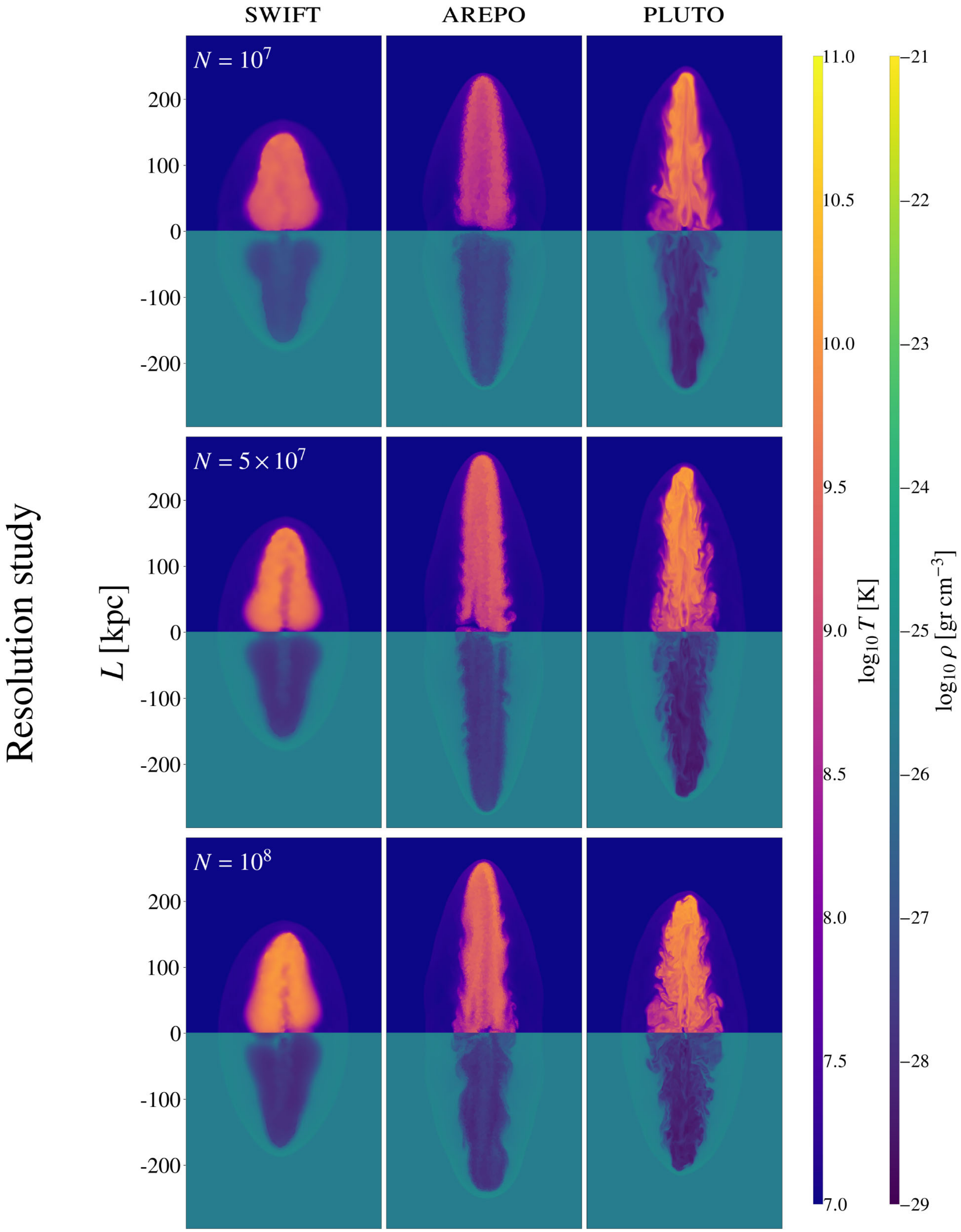

**Figure 8.** Temperature–density slices of jets simulated with the three codes in the uniform medium after 98 Myr, at three different resolutions ($N = 10^7$, $N = 5 \times 10^7$, $N = 10^8$), shown in separate rows. SWIFT lobes present only minor changes at different resolutions, while AREPO and PLUTO lobes significantly increase and then decrease their length with increasing resolution.

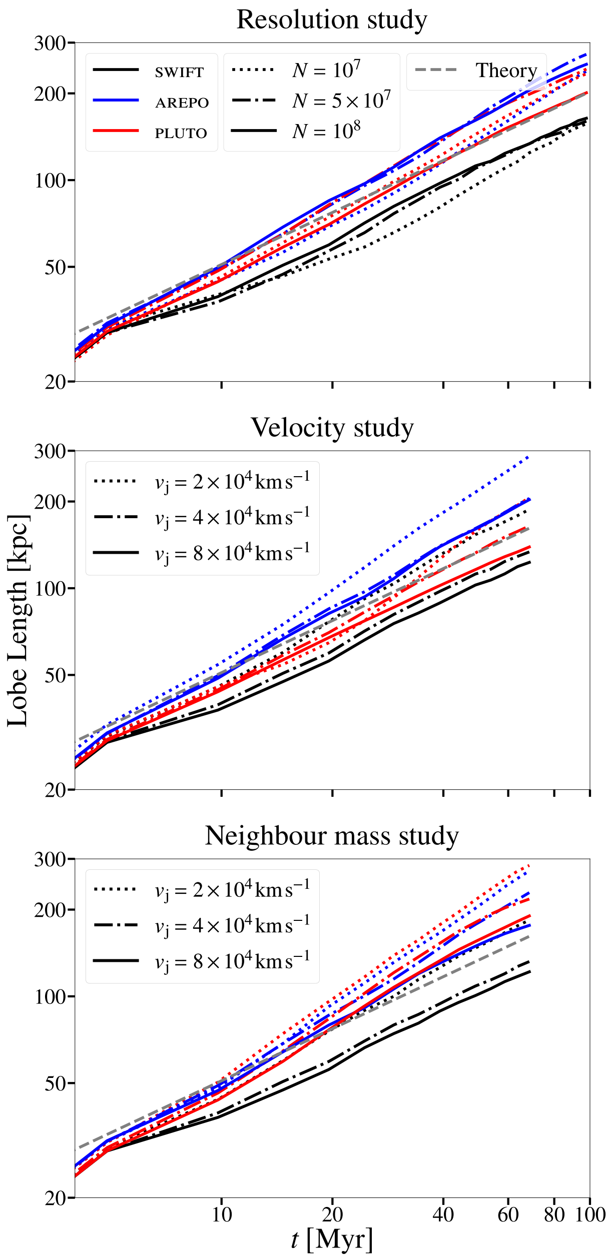

**Figure 9.** Lobe length as a function of time for all our runs. Coloured lines represent the three codes, line style shows the parameter variation, and the dashed grey line is the analytic prediction. **Top row:** Runs at three different resolutions. SWIFT lobes show only minor variations, while AREPO and PLUTO lobes become longer at intermediate resolution and shorter again at the highest resolution. **Middle row:** Runs with three different injection velocities. Fast-slow jets inflate shorter lobes that expand in closer agreement with the slope of the analytic solution, whereas slow-heavy jets produce longer lobes that tend to overpredict the analytic slope. **Bottom row:** Runs with three different injection velocities using the fixed-neighbour-mass scheme. Only PLUTO exhibits a systemic change relative to the fiducial scheme, now behaving very similarly to AREPO.

at higher resolutions[2]. We speculate that this could be another factor contributing to the decrease in lobe length observed at the highest resolution ($N = 10^8$) in these codes (P. M. Yates et al. 2018; R. Weinberger et al. 2023). An additional important feature, mainly observed in the SWIFT and AREPO lobes, is the development of stronger backflows with increased resolution. These backflows contribute to the fusing of the top and bottom lobes at their base, and could lead to the widening of lobes (F. Huško & C. G. Lacey 2023a).

Interestingly, we find very good consistency between resolutions regarding lobe masses and energetics (not shown here). The only noteworthy trend is a slight increase in the lobe thermal energy, accompanied by a decrease in the lobe kinetic energy with increasing resolution. This is also evident in the slightly higher lobe temperatures in the slices. A likely explanation is that the faster flows at higher resolutions can more effectively thermalise through shocks and adiabatic expansion work, raising the thermal energy at the expense of kinetic energy.

Encouragingly, for code-to-code consistency, the jets in the different codes appear more similar to each other in the lower-resolution runs. These jets have initial mass resolutions close to the resolutions used in many cosmological hydrodynamical simulations (although we achieve higher spatial resolutions and simulate more injection events). This suggests that AGN feedback, as implemented in cosmological simulations, may not be very different across different codes, or at least not until higher resolution can be achieved.

### 4.3 Effect of injection velocity

Jet velocity is a key parameter governing the propagation of jets. Observationally, FR-II lobes are powered by relativistic jets, while there is evidence that FR-I jets are slowed down on ∼ kpc scales (R. A. Laing & A. H. Bridle 2013), indicating a direct connection between jet speed and lobe morphology. In simulations, the behaviour of AGN jets can change radically with variation of the jet injection velocity (W. English et al. 2016; R. Weinberger et al. 2017; F. Huško et al. 2023; F. Huško & C. G. Lacey 2023a), suggesting that differences between codes can be highly dependent on the parameters used to model the AGN jets.

To demonstrate this, we compare the three codes in the uniform medium and at our highest resolution, with three different jet injection/virtual-particle velocities $v_\mathrm{j}$. We keep all other parameters fixed, barring a change to the virtual-particle mass, so as to maintain the same number of injection events. Keeping the jet power, and thus the injected energy per time the same means that higher velocity injections lead to faster, lighter jets, while lower velocity injections lead to slower, heavier jets (see equation 9). We evolve the jets for 68 Myr, ensuring that their lobes stay within the simulation boxes, and include runs with $v_\mathrm{j} = 2 \times 10^4\,\mathrm{km\,s^{-1}}$ ($m_\mathrm{j} = 1.6 \times 10^5\,\mathrm{M_\odot}$), $v_\mathrm{j} = 4 \times 10^4\,\mathrm{km\,s^{-1}}$ ($m_\mathrm{j} = 4 \times 10^4\,\mathrm{M_\odot}$), and $v_\mathrm{j} = 8 \times 10^4\,\mathrm{km\,s^{-1}}$ ($m_\mathrm{j} = 1 \times 10^4\,\mathrm{M_\odot}$).

Fig. 10 shows temperature-density slices of jets in the uniform medium at $t = 68$ Myr, produced by the three codes, with different injection velocities shown in separate rows. The simulations with $v_\mathrm{j} = 4 \times 10^4\,\mathrm{km\,s^{-1}}$, in the middle row, represent our fiducial runs. Fast and light jets produce lobes that are shorter, wider,

[2] Even though our simulated SWIFT lobes are mostly smooth, KH and RT instabilities are evident in higher-resolution SPH jet simulations (F. Huško & C. G. Lacey 2023a, b).

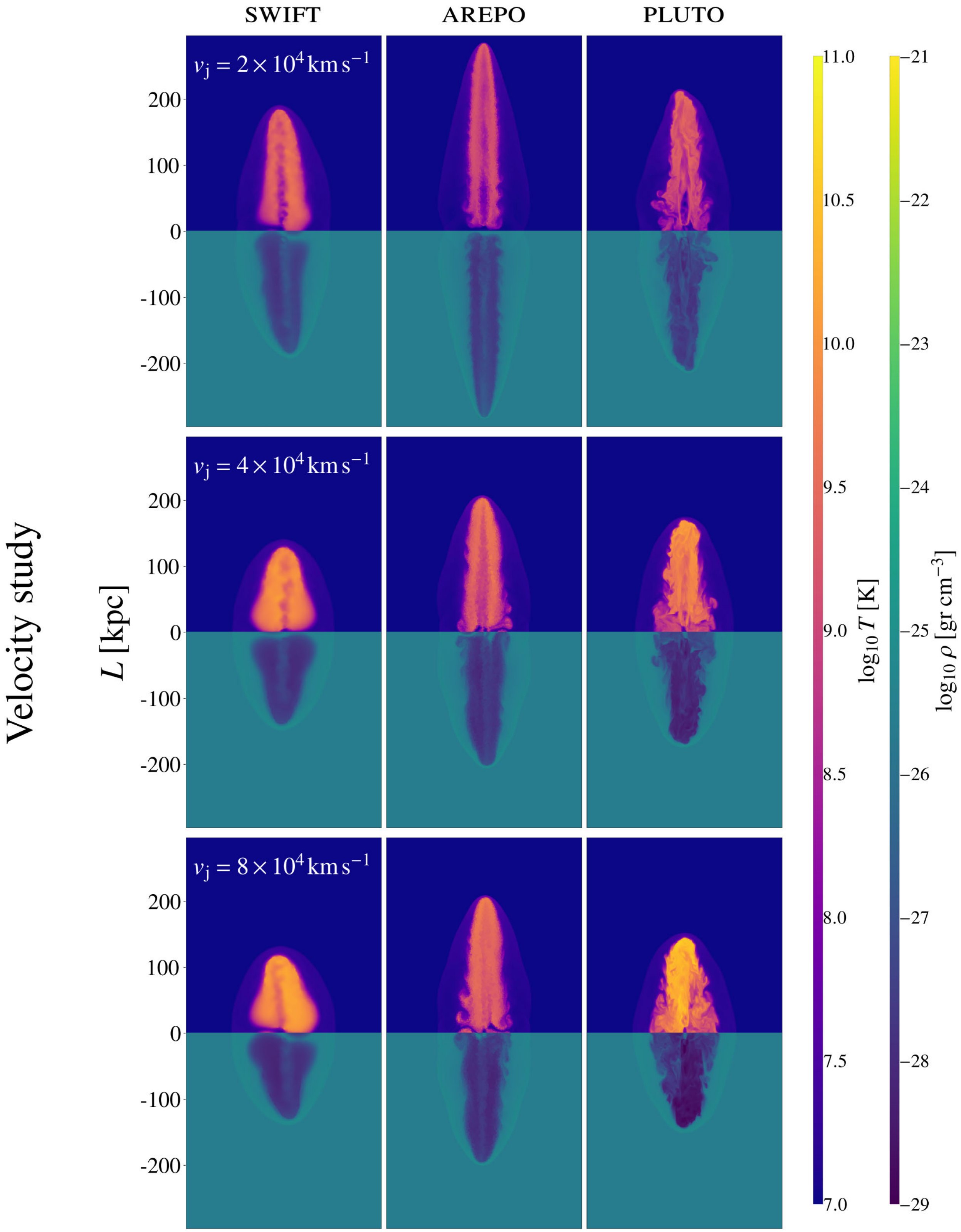

**Figure 10.** Temperature-density slices of jets simulated with the three codes in the uniform medium after 68 Myr, with three different jet injection velocities ($v_\mathrm{j} = 2 \times 10^4\,\mathrm{km\,s^{-1}}$, $v_\mathrm{j} = 4 \times 10^4\,\mathrm{km\,s^{-1}}$, $v_\mathrm{j} = 8 \times 10^4\,\mathrm{km\,s^{-1}}$), shown in separate rows. Increasing jet injection velocity results in shorter, wider and hotter lobes with more obvious backflows.

and hotter, with more obvious backflows. This occurs because higher injection velocities lead to a larger fraction of the injected energy being thermalized through stronger shocking and adiabatic expansion of the lobes. Simultaneously, the higher pressures at the jet head redirect the shocked jet material in a strong backflow. In contrast, slow and heavy jets have more inertia and higher momentum flux, and are thus able to drill through the ambient medium more effectively, inflating longer and thinner lobes.

These trends are evident in the middle row of Fig. 9, where we compare the lobe evolution with respect to time for simulations with the three injection velocities and the analytic self-similar solution. Slow-heavy jets produce lobes that deviate from the analytic self-similar solution and mostly overshoot the normalized prediction for lobe length. On the other hand, fast-light jets inflate lobes that expand self-similarly for longer and even underpredict the normalized solution. We also note a slight increase in lobe volume with increasing velocity, consistent with adiabatic expansion combined with strong backflows. In terms of energetics, the fraction of kinetic energy in the lobes decreases while the fraction of thermal energy increases with injection velocity, reflecting the greater thermalisation of the injected energy and the corresponding rise in lobe temperature.

Comparing the three codes, we find that some trends persist regardless of jet injection velocity; for example, AREPO consistently produces the longest lobes and SWIFT the shortest. Other trends change with jet injection velocity, as some lobe properties display code-specific dependence; for instance, the lobe mass in AREPO slightly increases while in PLUTO it slightly decreases with higher jet velocities. As a result, at the fastest injection velocity ($v_\mathrm{j} = 8 \times 10^4$ km s$^{-1}$), AREPO achieves the highest lobe mass, in contrast to our fiducial parameter runs ($v_\mathrm{j} = 4 \times 10^4$ km s$^{-1}$). This suggests that code differences can be dependent on the specific choices and parameters in a jet feedback scheme.

### 4.4 Neighbour mass injection

The results of our comparison study depend critically on the implementation of the subgrid jet model – specifically, on how the injected mass, momentum, and energy couple to the computational domain. In our fiducial model, this occurs via a virtual particle initiating a neighbour-finding loop, through which the nearest 10 gas elements receive feedback (see Section 3.2). In this set-up, the final state of the gas elements depends on their initial mass and, consequently, on the mass resolution of each code. SWIFT and AREPO use particle-splitting and mass refinement, respectively, so as to maintain an approximately constant mass resolution. In contrast, PLUTO cells can reach arbitrarily low densities, especially in regions directly affected by jet activity. As a result, when injecting over a fixed number of neighbours, the total neighbour mass remains roughly consistent in SWIFT and AREPO, but becomes significantly lower in PLUTO due to the lower density in the injection region. For a fixed injected energy, this leads to variations in the effective jet velocity and influences the subsequent propagation of the jet.

To illustrate this, we include here a controlled comparison of the three codes in the uniform medium and at our highest resolution, where the injection happens for a fixed neighbour mass: $M_\mathrm{ngb} = 20 \times 10^5\,\mathrm{M}_\odot$). We perform tests with three different jet injection velocities and evolve the jets for 68 Myr, following the same strategy as in Section 4.3. Fig. 11 shows temperature-density slices of jets in the uniform medium at $t = 68$ Myr, produced by the three codes, with different injection velocities shown in separate rows. For reference, we overlay the contours of the lobes in the case of the fiducial fixed-neighbour-number scheme (Fig. 10).

In SWIFT, the lobe geometry (and thermodynamic properties) are very similar in the two different injection schemes. This is primarily due to the fact that SPH particles can only strictly increase in mass, up to a factor of two, before being split. As a result, SWIFT is effective at preserving the mass of the resolution elements in the injection region, leading to minimal variation between the two injection schemes.

In the case of AREPO, we do see some slight deviations from the fiducial model, but we cannot find any systemic differences in lobe geometry, energetics or the development of instabilities and backflows. Thanks to its mass refinement algorithm, AREPO constrains most resolution elements to within a factor of 2 of their original mass. However, since mass can flow into or out of cells, and refinement is restrained by an additional geometric regularity criterion (making sure that very distorted cells do not get refined), some variation in cell masses is expected. As a caveat, we had to disable this extra geometric criterion for the lowest velocity ($v_\mathrm{j} = 2 \times 10^4$ km s$^{-1}$) (and highest mass loading) runs with this injection module to allow the code to converge to the target neighbour mass. The observed discrepancies between the two injection schemes are also dependent on the stochasticity of individual runs and sensitive to the generation of random angles in the jet launching module.

In contrast, PLUTO shows a consistent and systematic difference; the lobes are longer and cooler than in the fiducial injection scheme. This results from injecting into a (much) higher neighbour mass, which, for a fixed kinetic energy injection, leads to lower jet velocities. Reduced velocities result in weaker shocks, which deposit less thermal energy into the lobes, making them cooler. Moreover, lobes retain more kinetic energy and so preferentially expand along the jet axis rather than laterally.

Interestingly, in the fixed-neighbour-mass injection scheme, AREPO and PLUTO behave more similarly. This is evident in the bottom row of Fig. 9, where the evolution of lobe length with time shows the two codes exhibiting convergent behaviour across all injection velocities. The contrast is particularly clear when compared to the middle row, which shows results from the fixed-neighbour-number injection scheme, where AREPO produces consistently longer lobes than PLUTO.

This test highlights how differences in mass resolution between numerical methods can influence jet evolution, especially when combined with injection schemes where neighbour mass is not held fixed (see also M. A. Bourne et al. 2015). It underscores the need for careful prescription of injection modules when comparing simulations across codes with different resolution strategies.

### 4.5 Stratified medium jets

While most of our simulations were run in a uniform medium, realistic galaxy cluster atmospheres exhibit density gradients, with density decreasing from the cluster centre. Including such a set-up is particularly important for a code comparison project, as each code will achieve different initial mass and/or spatial resolution. Our assumed density profile is an isothermal beta profile, as described in Section 3.3. For this set-up, we run our simulations for 44 Myr to produce similar-length jets, code by code, as in the uniform-density medium.

The initial conditions for SWIFT and AREPO are generated by sampling resolution elements using inverse transform sampling (L. Devroye 1986) inside a sphere, and then retaining those that

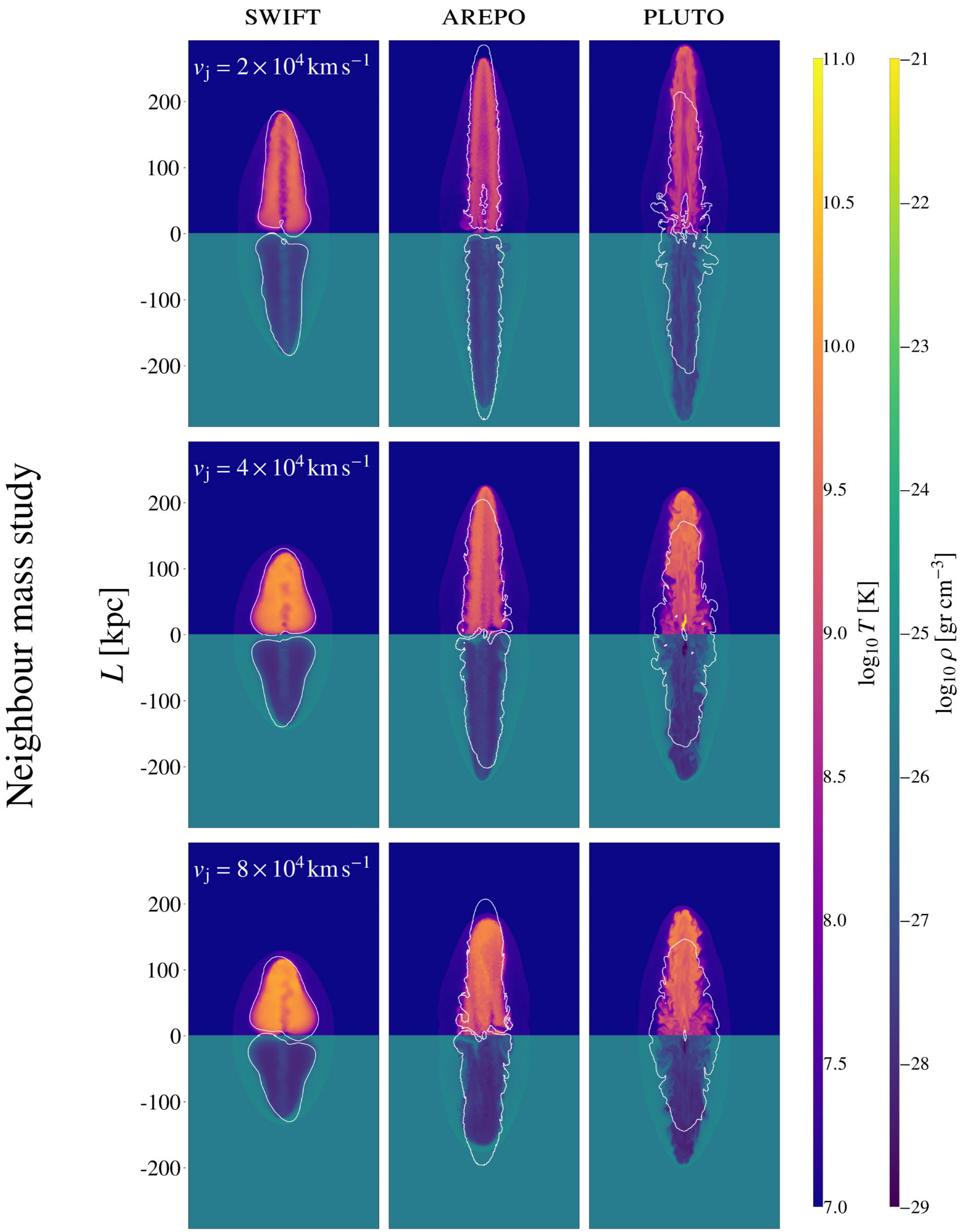

**Figure 11.** Temperature–density slices of jets simulated with the three codes in the uniform medium after 68 Myr, with injection into the same neighbour mass. We test three different jet velocities, shown in separate rows, and include the lobe contours of the fiducial fixed-number-of-neighbours scheme. SWIFT lobes are almost identical between the two schemes. AREPO lobes present some deviations but no systemic differences. PLUTO lobes are consistently longer and cooler.

reside inside our box for a total of $N = 2.9 \times 10^7$ ($M_{\rm gas} = 10^5\,{\rm M}_\odot$) elements. We test that the initial gas distribution remains stable, in equilibrium with the external gravitational potential, for at least four times the duration of the jet episode. Additionally, we let the system relax adiabatically for $\sim 200$ Myr, before launching the jet to smooth out any initial over/under densities. In the case of PLUTO, its static grid allows the analytic density profile to be directly applied in the initial configuration.

It is important to note that these two distinct set-ups – sampling equal-mass elements for SWIFT and AREPO, and applying an analytic density model on a grid for PLUTO – affect effective resolutions in different ways. SWIFT and AREPO will start the simulation with more resolution elements, of a fixed mass, in the densest regions, i.e. the centre of the domain, with progressively fewer elements as the density falls off with radius. The initial spatial resolution in these codes will therefore decrease with radius. In contrast, PLUTO uses the same spatial resolution in all of the domain, but its mass mass resolution varies to follow the density profile. This makes it harder to achieve a fair comparison; we find that PLUTO consistently exhibits lower effective resolution in this set-up if we use the same number of resolution elements. To work around this, we include a PLUTO simulation with the same number of grid cells as initial resolution elements in SWIFT and AREPO (standard resolution: SR) and another one where the grid spatial resolution matches the mean spatial resolution inside the density profile core ($r < 10$ kpc) in the SWIFT and AREPO initial conditions (high resolution: HR). We achieve the latter by setting the cell volumes in PLUTO to be the same as the mean cell volume in the core of the relaxed AREPO initial conditions.

In the top panel of Fig. 12, we show temperature-density slices of jets simulated with the three codes in the stratified medium at $t = 44$ Myr. We note that our stratified medium reaches lower densities than the uniform medium beyond $\sim 32$ kpc. Hence, all of the lobes appear longer and thinner than in the uniform medium, remain clearly separated and do not develop the strong backflows we observed in the previous sections. We find a more significant departure from the analytic self-similar solution (especially for AREPO), with the lobes increasing their aspect ratio throughout the run. Moreover, we note less lobe deceleration but more consistent lobe volumes and masses across the codes. Additionally, we notice a larger portion of the lobe energy being in kinetic form in all the codes, with AREPO specifically, retaining more overall energy in the lobes. These trends reflect the jets propagating into progressively lower-density regions, encountering less external ram pressure at the jet head and advancing more freely.

Comparing the three codes, we observe very similar trends to those in the uniform-medium runs: SWIFT inflates short and wide lobes; AREPO produces long and thin lobes; and the SR of PLUTO produces intermediate-length lobes. Additionally, the HR resolution lobes are shorter, with more pronounced KH instabilities, following the trend we observed in the resolution study in the uniform medium in Section 4.2.

In the bottom panel of Fig. 12, we show the ambient medium profiles for density (top left), pressure (top right), temperature (bottom left), and entropy (bottom right). The density and temperature profiles represent volume and mass weighted averages, respectively, for different radial bins, while the pressure and entropy curves are derived afterwards as $P = (\gamma - 1)\, u\, \rho$, $S = T\, \rho^{-2/3}$, where $S$ is a measure of entropy that is monotonically related to the thermodynamic entropy. We make sure to exclude the lobes when calculating the profiles to isolate the impact of the jets on the ambient medium. The dashed line represents the initial conditions, black the SWIFT simulation, blue the AREPO and red the HR PLUTO runs. The SR PLUTO run presents only minor differences from the HR PLUTO run.

All simulations exhibit clear deviations from hydrostatic equilibrium within the inner 100–200 kpc, as the jets displace and heat the ambient medium. However, care must be taken when interpreting the innermost 10 kpc, indicated by the shaded region, where direct injections of feedback may influence the results. Moreover, since the jets expand along a single axis, the effects seen in the spherically averaged profiles are diluted. This will be especially relevant for AREPO jets which are more elongated and thus fill a smaller solid angle.

Despite differences in morphology, all jets have a broadly similar impact on the ambient medium: they heat the gas in the inner regions and displace it to larger radii. This results in elevated mean gas temperature and entropy along with a rise in density and pressure, followed by a drop as the jets propagate outward. SWIFT and PLUTO jets have a more pronounced effect on gas density and pressure at intermediate radii ($r \sim 50$ kpc). while PLUTO jets, in particular, produce the strongest temperature and entropy spikes at small radii ($r < 20,$ kpc). In contrast, the longer AREPO jets extend their influence to larger radii, leaving noticeable imprints well beyond the core. This suggests that shorter jets with wider and hotter lobes are more efficient at heating and uplifting gas at small radii, potentially more directly influencing gas with the shortest cooling times.

### 4.6 Remnants

As AGN jets propagate through the ambient medium, they inflate hot, low-density lobes that expand outward, displacing the dense central regions of the stratified atmosphere. This process generates shocks and sound waves and drives bulk motions and turbulence into the surrounding gas, potentially decreasing or disrupting the large-scale accretion flow onto the central black hole powering the jets (M. A. Bourne & H.-Y. K. Yang 2023). Once the jets shut off, the lobes no longer receive a fresh supply of momentum and energy and begin to rise buoyantly through the stratified atmosphere (G. S. C. Stewart et al. 2025). Candidates for such AGN remnants have been identified in observations (e.g. P. Parma. et al. 2007; M. Murgia et al. 2011; L. E. H. Godfrey, R. Morganti & M. Brienza 2017; V. H. Mahatma et al. 2018; S. Dutta et al. 2023), based on the balance of accretion and feedback (N. Jurlin et al. 2020; S. S. Shabala et al. 2020), and have been studied in simulations under the influence of cluster weather (S. Heinz et al. 2006; B. J. Morsony et al. 2010; M. A. Bourne et al. 2019; M. A. Bourne & D. Sijacki 2021).

To study the properties of jet remnants, we use a larger box with a volume of 900 × 900 × 1200 kpc$^3$. We use the stratified medium to allow for the buoyancy of the hot remnants. We set up our initial conditions as described in the previous section, this time with $N = 1.8 \times 10^7$ resolution elements ($M_{\rm gas} = 10^6\,{\rm M}_\odot$). We turn on the jets for 44 Myr, and subsequently allow the remnants to evolve during the quiescent stage for another 152 Myr (for a total of 196 Myr).

In the top panel of Fig. 13, we show 3D volume renderings of the lobes of our jet remnants, while in the middle panel we present temperature–density slices of the remnants and the stratified medium, at $t = 196$ Myr. The physical evolution of these remnants depends on the state of the jets at the end of the

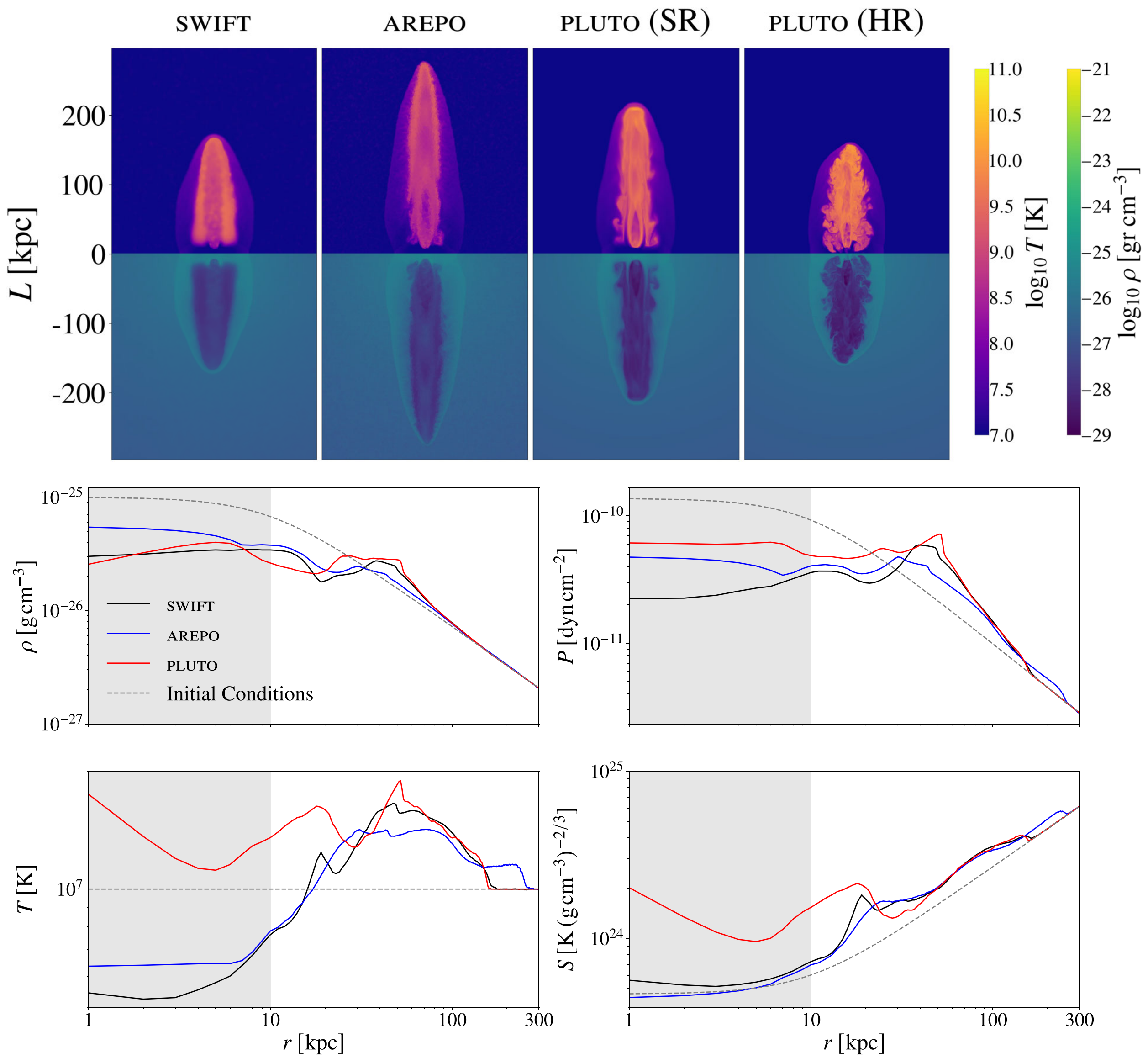

**Figure 12.** Top: Temperature-density slices of SWIFT, AREPO, and PLUTO jets in a stratified medium after 44 Myr. We include two resolution cases for PLUTO for a fairer comparison; SR denotes standard resolution and HR high resolution as described in the main text. All jet lobes appear longer and thinner compared to the uniform medium lobes and the analytic self-similar solution for the stratified medium. Bottom: Ambient medium profiles for density (top left), pressure (top right), temperature (bottom left), and entropy (bottom left). The dashed line represents the initial conditions, black the SWIFT simulation, blue the AREPO and red the HR PLUTO. All jets increase gas temperature and entropy and cause a rise in density and pressure followed by a drop in the inner regions.

active jet phase, as well as their interaction with the surrounding medium and gravitational field.

SWIFT jets evolve into stubby, smooth, and cylindrical bubbles with limited small-scale structure. In contrast, AREPO jets produce long, thin, and filamentary remnants that exhibit more internal mixing. The SR PLUTO simulation produces intermediate-length remnants where much of their interior material has already mixed into the ambient medium. This mixing, also apparent, although to a lesser degree, in the AREPO remnants, is the result of the development and growth of a large-scale Rayleigh–Taylor (RT) instability. This instability develops because the gravitational field accelerates denser ambient gas into the lighter lobe material. Since the lobe is embedded in the ambient gas, the RT instability manifests as circular overturning motions that start to mix the lobes from the base and progress inward/upward. These motions can be coupled to the growth of KH modes that further enhance mixing (C. S. Reynolds et al. 2005; F. Huško & C. G. Lacey 2023b). By contrast, the HR PLUTO remnants remain more coherent, taking on a wider cylindrical shape, though signs of mixing are also apparent, with some trailing gas and prominent KH instabilities.

In Fig. 14, we display temperature-density slices for the three jets and remnants, for progressively later times in each row. The first row corresponds to the time of jet turn-off and is a lower resolution version of the top panel of Fig. 12. In general, the properties of jets and their lobes at turn-off appear to play an important role in shaping the differences between remnants of different codes. The relatively smooth and short SWIFT lobes of the active phase

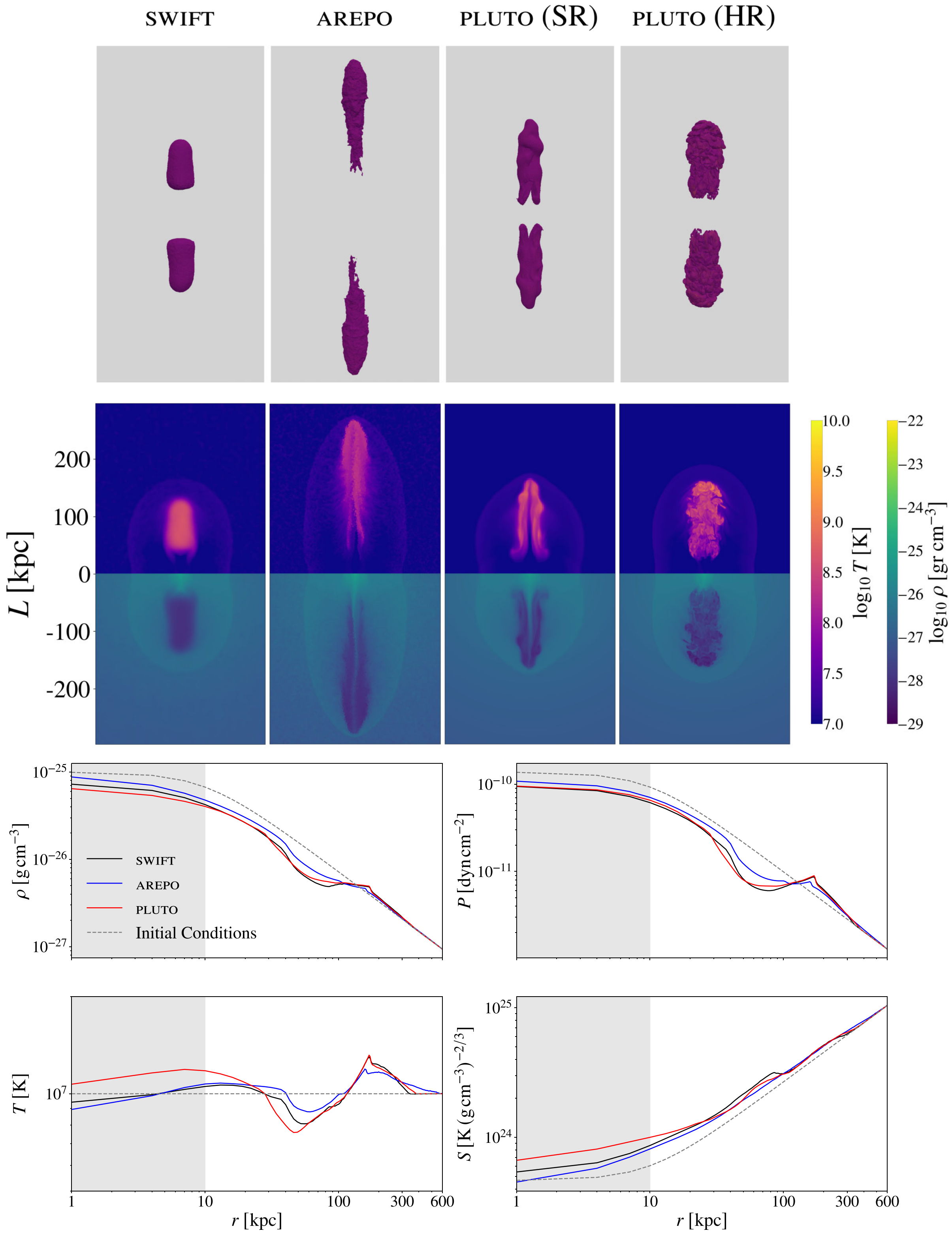

**Figure 13.** Top: 3D Volume renderings of remnant lobes simulated with the three codes in the stratified medium, after 196 Myr (when jets turn off after 44 Myr). Middle: Temperature-density slices of jet remnants and their ambient medium. SWIFT produces cylindrical bubbles while AREPO long filamentary remnants. PLUTO SR presents a thin hollow remnant with internal mixing, while its HR run produces a more coherent structure with instabilities and trailing gas. Bottom: Ambient medium profiles after 196 Myr. The profiles have relaxed closer to their new equilibrium in the absence of cooling. All jets and remnants have produced similar effects at this stage.

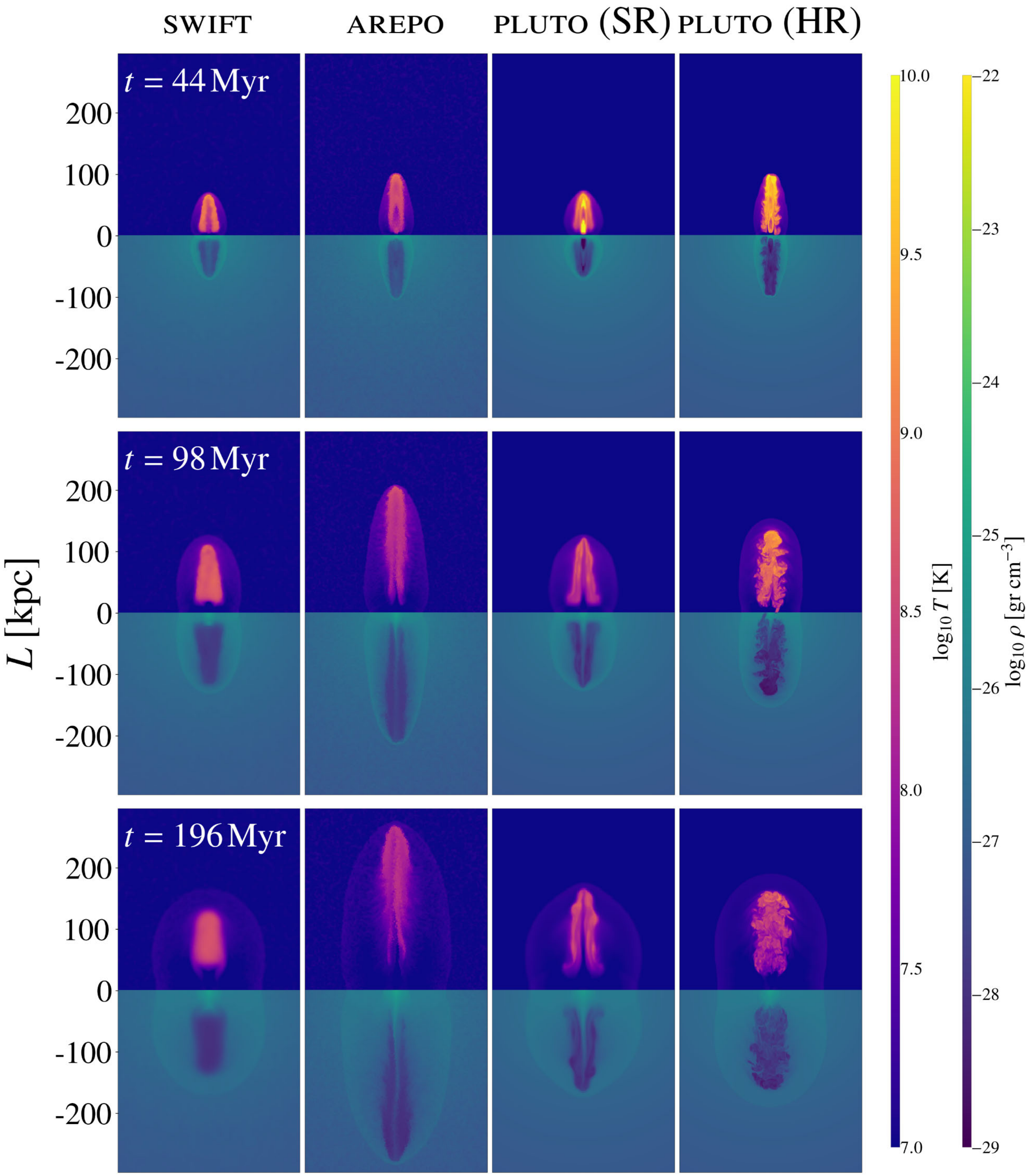

**Figure 14.** Temperature–density slices of the time evolution of the jets and their remnants simulated with the three codes in the stratified medium. Different rows showcase the different jets and remnants at progressively later times, with the jets turning off at $t = 44$ Myr.

evolve into stubby, cylindrical remnants, while the long, thin, and cool AREPO lobes develop into elongated, cool remnants. The two PLUTO runs start with very hot lobes that eventually become cooler via mixing and adiabatic expansion as the jets turn off and the lobes start expanding and rising in the stratified medium. The KH instabilities visible in the HR remnant can also be seen in the lobe during the active jet phase. We find that AREPO jets produce highly elongated remnants even though their active lobes seem to be as long as the HR PLUTO lobes, which instead develop more cylindrical remnants. This difference likely arises from the thinness of the AREPO lobes and their higher kinetic energy content. By contrast, the HR PLUTO lobes convert a larger fraction of their energy in thermal form, as evidenced by the higher lobe temperatures, which leads to a more isotropic expansion.

Rising buoyantly through the stratified medium, these hot, underdense remnants are slowed by drag forces and shaped by the presence of the wake and drift (E. C. D. Pope et al. 2010; F. Huško & C. G. Lacey 2023b), ultimately attaining their terminal velocity after approximately 75–125 Myr. Across all codes, the remnants gradually reach remarkably similar terminal velocities and lobe masses. Over the course of the remnant phase, most of the energy in the lobes is transferred to the ambient medium. By the end of the run, less than 10 per cent of the injected energy remains as thermal and kinetic energy in the lobes, while around 70–80 per cent ends up in the medium, primarily as thermal energy. Moreover, a significant fraction of the injected energy, about 20 per cent, contributes to the gravitational potential energy of the displaced gas. These results are broadly consistent between SWIFT, AREPO and both resolutions of PLUTO and are not particularly influenced by our definition of the lobe.

In the bottom panel of Fig. 13, we plot the ambient medium profiles for density, temperature, pressure and entropy after 196 Myr. As expected, once the jets shut off, the profiles begin to move closer to their new equilibrium under the adiabatic conditions of our simulations. Even without cooling, some residual effects of jet activity still remain. Both density and pressure remain slightly suppressed at small to intermediate radii, while the entropy profile shows a clear enhancement at intermediate radii, consistent with irreversible heating caused by earlier jet-driven shocks. Notably, all three codes predict broadly similar profiles, with only minor variations.

## 5 DISCUSSION

### 5.1 Comparison with previous jet simulations

Our code comparison project builds upon a wealth of previous high-resolution AGN jet simulations (e.g. M. J. Hardcastle & M. G. H. Krause 2013; W. English et al. 2016; M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017, 2023; P. M. Yates et al. 2018; F. Huško & C. G. Lacey 2023a, b). Comparing our new jet model (see Section 3.2) with select implementations from the literature can prove useful in understanding how differences in jet launching and injection schemes influence jet propagation, lobe inflation, and the deposition of feedback energy in the ambient medium. This can also shed light on how individual AGN feedback schemes can be effectively designed and honed for specific codes.

#### 5.1.1 *SWIFT jets*

F. Huško & C. G. Lacey (2023a) developed a novel jet injection technique within the SWIFT code that employs the SPH method. The jets were implemented as a series of discrete injection events, realized by shooting hydrodynamically decoupled SPH particles and recoupling them on scales of a few kpc, once they cleared the central jet-launching region. The key difference from our model lies in the treatment of decoupled particles: rather than using virtual particles and distributing their mass, momentum, and energy to existing neighbouring particles (as done in our model), their scheme directly hydrodynamically recouples SPH particles to represent injection events. While both approaches can effectively model jet launching and lobe inflation, their method introduces additional SPH particles into the lobes, while ours increases the masses of existing particles (up to twice the initial value), thereby slightly reducing the effective resolution in the lobes. Nonetheless, direct comparison of the methods is difficult, as the different injection techniques lead to distinct couplings between the jet parameters; for instance, the (effective) jet injection velocity in their model is set as an independent parameter, whereas in ours it depends on neighbour number/mass and initial velocities as well.

#### 5.1.2 *AREPO jets*

The flexible refinement techniques in AREPO have enabled the development of various high-resolution jet models. In M. A. Bourne & D. Sijacki (2017), jets were assumed to be pre-collimated and were launched by continuously injecting mass, momentum, and energy into existing gas cells within a small, highly refined cylinder along the jet direction. The authors employed multiple refinement techniques to better resolve both the injection region and the jet lobes by using jet tracer- and gradient-dependent refinement criteria. This allowed for more computationally efficient simulations in which lobe structure and instabilities were captured at higher resolution. Such refinement criteria were demonstrated to be critical to model continuously injected, light jets, whereas pure Lagrangian refinement led to poorly resolved and/or stunted jets.

An alternative approach was presented in R. Weinberger et al. (2017), where the authors set up (magneto) hydrodynamical simulations of jets and remnants in AREPO. In this case, the jets were injected by setting the exact thermodynamic state (density and internal energy) and gas velocity within two regions, close to the centre of the domain and along the jet direction. Similar to M. A. Bourne & D. Sijacki (2017), the jet was assumed to be pre-collimated. They also included jet tracer- and volume-dependent refinement to override the standard mass refinement to achieve better resolution in the lobes, without which the jets and lobes were poorly resolved.

The above studies highlight the importance of refinement in moving-mesh simulations, especially in simulating fast, light jets that inflate low-density lobes. These have been supplemented by a range of other works that consider jets in realistic cluster environments (M. A. Bourne et al. 2019; M. A. Bourne & D. Sijacki 2021), the impact of cosmic rays and/or magnetic fields (K. Ehlert et al. 2018), and self-regulated feedback in clusters (K. Ehlert et al. 2022) or on smaller scales assuming spin-driven jets (R. Y. Talbot et al. 2021; R. Y. Talbot, D. Sijacki & M. A. Bourne 2022, 2024). Specialized refinement is crucial in jet models that include a continuous injection at low powers and/or light jets, where insufficient resolution could otherwise lead to excessive numerical mixing. We expect our discrete feedback scheme to mitigate this issue, even with mass refinement alone, though the resolution of the lobes will still be affected. Nonetheless, low-resolution simulations remain valuable as refinement techniques can be very computationally expensive over long timescales.

#### 5.1.3 *PLUTO jets*

Most simulations of AGN feedback with PLUTO (see e.g. M. J. Hardcastle & M. G. H. Krause 2013, 2014; W. English et al. 2016, 2019; P. M. Yates-Jones et al. 2021, 2023) inject jets continuously by applying inflow boundary conditions and, in some cases, overwriting the hydrodynamic values (density, pressure, velocity) of the grid cells that define the injection region. This approach leads to more stable and reproducible results (than launching virtual particles at random angles), but, at least in the latter case, it has the drawback of not self-consistently evolving the central region. PLUTO jet simulations almost always employ the special relativistic module (RHD) in the code to capture relativistic effects, and sometimes even the magnetohydrodynamics module (RMHD) alongside it, to self-consistently evolve magnetic fields. As mentioned earlier, magnetic fields and relativistic effects were not included in our simulations, in order to facilitate a fair and clear hydrodynamics comparison of codes.

It is worth mentioning that some studies of astrophysical jets with PLUTO (e.g. G. Mattia et al. 2023) use the HLL Riemann solver instead of the more standard HLLC (or HLLD). The HLL solver is more diffusive but also more stable, which can prove advantageous in simulations of jets. Our main suite of PLUTO simulations was run with the more accurate HLLC (while switching to the HLL in the presence of strong shocks), but we additionally tested the HLL solver. In general, most of the trends (see Section 5.1.4) and code differences remained unchanged, although the jets and lobes tended to be slightly longer and exhibited weaker KH instabilities.

### *5.1.4 General behaviour*

Across all simulation codes and different studies, we observe several consistent trends in AGN jet behaviour:

(i) **Self-similar evolution:** Certain jet parameters can lead to large-scale jet and lobe evolution that tends to self-similarity. In our uniform-medium runs, higher injection velocities led to more consistent self-similar behaviour in all the codes. In the stratified medium, the jets diverged further from the analytic predictions, inflating more elongated lobes with growing aspect ratios. This is in line with previous high-resolution simulations (M. J. Hardcastle & M. G. H. Krause 2013; W. English et al. 2016; F. Huško & C. G. Lacey 2023a).

(ii) **Jet velocity:** Assuming equal powers, fast-light jets tend to be shorter and inflate wider, hotter lobes, as more of their kinetic energy can be thermalized through shocks and adiabatic expansion ($P\,\mathrm{d}V$) work in the ambient medium. In contrast, slow-heavy jets propagate further and produce thinner, cooler lobes, due to their higher inertia. This trend was evident in our results and has been noted in previous studies (W. English et al. 2016; R. Weinberger et al. 2017; F. Huško et al. 2023; F. Huško & C. G. Lacey 2023a).

(iii) **Lobe energetics:** Jet lobes typically retain around 40–50 per cent of the injected energy (M. J. Hardcastle & M. G. H. Krause 2013; W. English et al. 2016; M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017; M. A. Bourne et al. 2019; M. A. Bourne & D. Sijacki 2021) – although some studies find slightly lower fractions (F. Huško & C. G. Lacey 2023a) and others report significantly lower values when including relativistic effects (M. Perucho et al. 2017). In our simulations, most cases were consistent with around 40–45 per cent energy retained in the lobes by the end of the run. The partition of this energy (thermal or kinetic) depends on the jet model, parameters, and environment. In general, fast-light jets convert more of their energy into thermal energy than slow-heavy jets, due to stronger shocks and increased $P\,\mathrm{d}V$ work.

(iv) **Jet length and resolution:** Jet length is a complex function of resolution. Lower resolution causes artificial SPH smoothing in particle codes and numerical mixing and diffusion in grid codes. In general, this results in shorter jets (M. A. Bourne & D. Sijacki 2017; R. Weinberger et al. 2017, 2023; F. Huško & C. G. Lacey 2023a). This effect was noted in our own simulations, especially for the AREPO jets. Moreover, it was particularly noticeable for all of the codes at low injection velocities ($v_\mathrm{j} < 2 \times 10^4\,\mathrm{km\,s^{-1}}$). In resolution tests not shown here, such jets evolved ballistically – maintaining nearly constant bulk lobe speed – and became significantly longer as resolution increased. A similar trend was observed in F. Huško & C. G. Lacey (2023a). However, at very high resolutions, Kelvin–Helmholtz instabilities may become better resolved, potentially reducing jet length again (P. M. Yates et al. 2018; R. Weinberger et al. 2023; F. Huško & C. G. Lacey 2023a). We saw this in the PLUTO jets in both the uniform and the stratified medium. Still, interpreting these instabilities requires care (M. A. Bourne & D. Sijacki 2017). Overall, jet length likely increases with resolution up to a point, then may decrease once again, as physical instabilities are better resolved.

(v) **Instabilities in remnant bubbles:** The hot, low-density remnant bubbles left behind by jets can develop RT and KH instabilities as they rise through denser ambient gas. These instabilities lead to increased mixing and potential dissolution of the bubbles. A large-scale RT instability was observed in our AREPO and even more so in our SR PLUTO remnants, while KH instabilities were seen in the HR run. Their effects depend strongly on resolution, initial conditions, and relevant timescales, and may even be suppressed in more realistic set-ups (C. S. Reynolds et al. 2005; R. Weinberger et al. 2017; W. English et al. 2019; F. Huško & C. G. Lacey 2023b).

### *5.1.5 Numerical effects*

There is now growing evidence that different numerical solvers or codes can lead to differences in jet propagation, bubble inflation and regulation of cooling in clusters. In D. Martizzi et al. (2018), the authors compared the HLLC and HLLE Riemann solvers of the grid code ATHENA, concluding that the former could better resolve the cold, overdense gas filaments that formed as a result of the interaction of jet propagation and cooling. Similarly, F. Huško & C. G. Lacey (2023b) showed that remnant bubbles simulated with minimal SPH methods differed in appearance and temperature from those simulated with modern SPH methods that include artificial viscosity limiters and artificial conduction. Additionally, G. Ogiya et al. (2018) found that AGN inflated bubbles are unstable to Kelvin–Helmholtz instabilities when modelled with grid, meshless codes or modern SPH methods, but that a minimal SPH model could result in bubbles that do not mix efficiently and remain well-formed, due to artificial surface tension. Finally, in R. Weinberger et al. (2023), the authors compared two codes (AREPO and GIZMO) in modelling AGN jet feedback. They used a different jet scheme and cooling function for each code and found that AREPO jets, compared to GIZMO jets, were more efficient at delaying cooling flows in clusters (though this is also model dependent). These studies illustrate how differences in numerical hydrodynamics propagate into the deposition of mass, momentum and energy in astrophysical systems, potentially influencing the interplay of feedback processes, cooling flows and accretion. Therefore, careful comparison of different AGN models and different codes as applied in such systems can help elucidate the actual microphysical processes at play and discover robust ways to model them.

### 5.2 Areas for further development

In this work, we have focused on idealized, purely hydrodynamical AGN jet simulations in simplified environments. While these clean set-ups allow for careful comparison of the hydrodynamics of the different jets, we necessarily sacrificed complexity in different aspects of modern jet modelling (see e.g. M. A. Bourne & H.-Y. K. Yang 2023, for a recent review).

As emphasized earlier, our study compares three different codes using the same jet model, rather than implementing subgrid feedback schemes tailored to each code. In recent years, hydrodynamical codes are increasingly being used in tandem with feedback models specific to their numerical framework, especially in more complex, high-resolution simulations, like zoom-in simulations of individual clusters. In future work, we aim to extend our comparison project by including feedback models tuned to each code and adopting set-ups that are more reflective of realistic astrophysical environments.

In galaxy formation simulations, AGN feedback is usually coupled to the supermassive black hole accretion rate to determine its power. Incorporating this coupling in our comparison framework would offer insights into how differences in jet propagation and deposition of momentum and energy at small radii influence the self-regulated cycle of black hole accretion and jet launching. Another feature of cosmological simulations is their use of additional physics modules, such as radiative cooling, star formation and magnetic fields, which can significantly impact jet evolution and lobe morphology. Including such modules would allow for more realistic modelling of AGN feedback and could highlight more subtle differences between codes. Crucially, cosmological environments include complex, evolving structures, characterized by inhomogeneities and turbulence, which are fundamentally different from idealized set-ups, impact jet evolution (e.g. S. Heinz et al. 2006; B. J. Morsony et al. 2010; M. A. Bourne et al. 2019; M. A. Bourne & D. Sijacki 2021; P. M. Yates-Jones et al. 2023) and must be taken into account for a self-consistent galaxy formation perspective.

We should also note that our comparisons were focused on three widely used hydrodynamical codes, spanning the spectrum from fully Lagrangian (SWIFT), to moving-mesh (AREPO), to Eulerian grid-based (PLUTO) methods. However, these represent a subset of a wealth of modern codes that are frequently used in astrophysical simulations (e.g. Teyssier, R. 2002; V. Springel 2005; H. Menon et al. 2015; V. Springel et al. 2021). Notably, we have not included the meshless finite-volume/finite-mass method (GIZMO) developed by P. F. Hopkins (2015). Given the competitiveness of the meshless methods in standard test problems and the development of the particle spawning jet model (K.-Y. Su et al. 2021), it would be valuable to include in comparisons of AGN jet models in the future.

Finally, we have not considered comparisons of the speed and efficiency of the different codes, as we were focusing on the physical properties and evolution of jets and the lobes that they inflate. We did not attempt to optimise our set-ups on these aspects and thus could have introduced unnecessary biases or inefficiencies that do not reflect the true performance of the methods. We reserve such an analysis for future work.

## 6 CONCLUSIONS

We performed simulations of idealized AGN jets using a novel launching module with three widely used astrophysical hydrodynamical codes, each representing one of the three main classes of computational fluid codes: SWIFT (SPH), AREPO (moving-mesh) and PLUTO (static grid).

(i) We simulated jets in a uniform medium, using identical parameters and initial conditions. All jets inflated lobes of hot gas that followed a self-similar scaling. SWIFT generated the shortest jets, with wide and hot lobes. AREPO produced the longest jets that formed the thinnest and coolest lobes. PLUTO jets and lobes displayed intermediate characteristics.

(ii) We tested three different jet injection velocities. Fast-light jets were more efficient at shock heating and adiabatic expansion and produced shorter, hotter lobes with stronger backflows. In contrast, slow-heavy jets had more inertia and higher momentum flux and thus produced longer, cooler lobes.

(iii) In a stratified medium with an external gravitational field, all jets diverged from the self-similar scaling and inflated longer and thinner lobes due to lower external ram pressure. The relative differences between codes closely mirrored those in the uniform case. Despite variations, all jets and lobes had a comparable impact on the ambient medium.

(iv) After the jets were switched off, SWIFT produced smooth cylindrical bubbles, AREPO formed long, thin filaments, and PLUTO jets evolved into intermediate-length remnants with fluid instabilities. In the absence of cooling, ambient gas profiles relaxed close to a similar new equilibrium across all codes.

(v) Code differences depended on how feedback coupled to the computational domain and the effective mass and spatial resolution of each method.

(a) The constant spatial resolution in PLUTO enabled very well-resolved flows within the lobes, showing clear signs of fluid instabilities.

(b) The adaptive spatial resolution in SWIFT and AREPO yielded lower resolution in the low density lobes (but higher near the bow shock). The lack of mixing and the kernel-based smoothing in SWIFT reduced the effective spatial resolution in the lobes more so than the moving-mesh in AREPO. Thus, SWIFT lobes remained smooth while AREPO lobes displayed some fluid instabilities.

(c) Our comparison of equal numbers of particles and cells disadvantaged SWIFT, since the kernel based smoothing in SPH inherently results in lower spatial resolution compared to other codes.

(d) In our fiducial fixed-neighbour-number injection scheme, PLUTO (with adaptive mass resolution) achieved higher effective jet velocities than AREPO (with constant mass resolution). Using a fixed-neighbour-mass scheme reduced differences between these two codes.

All in all, our results indicate that, with identical injection models and initial conditions, high-resolution simulations of AGN jets and remnants can still exhibit notable differences depending on the choice of hydrodynamical code.

While such differences are important for high-resolution, detailed studies of jet and lobe evolution, they may be less important in the context of cosmological simulations of structure formation, where the main focus is on capturing the impact of jet feedback on resolvable scales. Owing to resolution limitations, these simulations have traditionally focused on reproducing the global statistics of galaxy populations, rather than resolving the small-scale physics of feedback. In this regime, simplified AGN feedback models continue to be widespread, and the principal uncertainties stem from model selection, parameter tuning, and the interactions with prescriptions in other modules, while the specific treatment of hydrodynamics is likely less important.

However, modern advances in numerical modelling are enabling a gradual shift in this paradigm. Recent developments in injection methods and refinement strategies across different codes have brought to light novel ways for simulating jets across a variety of scales and environments. While resolution constraints in structure formation simulations still require simplified AGN feedback models, these are now increasingly being supplemented with advanced numerical methods that provide a stronger link between physical processes and numerical implementation (see e.g. M. Guo et al. 2025). It is therefore prudent to test the consistency of subgrid models across different hydrodynamical solvers; this is best achieved with high-resolution simulations, as employed in this work, which can serve as a guide to inform the prescriptions in simplified models. This approach not only helps qualify the robustness of AGN feedback models but also facilitates the development and calibration of future jet subgrid prescriptions. These prescriptions should deliver consistent results across different hydrodynamical codes and potentially approach the fidelity of high-resolution jet simulations.

## ACKNOWLEDGEMENTS

We thank the anonymous referee for their careful reading of the paper and their thoughtful comments. MAB acknowledges support from a UK Research and Innovation (UKRI) Stephen Hawking Fellowship (EP/X04257X/1) as well as the Science and Technology Facilities Council (STFC), grant code ST/W000997/1. CP acknowledges the support of the ARC Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D), through project number CE170100013. This research was undertaken with the assistance of resources from the National Computational Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government. This work was supported by resources provided by the Pawsey Supercomputing Research Centre's Setonix Supercomputer (https://doi.org/10.48569/18sb-8s43), with funding from the Australian Government and the Government of Western Australia.

*Software citations*: We thank the SWIFT, AREPO and PLUTO development teams for making these codes public. This research also relies heavily on the PYTHON (G. Rossum 1995) open source community, in particular, NUMPY (C. R. Harris et al. 2020), MATPLOTLIB (J. D. Hunter 2007), SCIPY (P. Virtanen et al. 2020), SWIFTSIMIO (J. Borrow & A. Borrisov 2020) and PYVISTA (C. B. Sullivan & A. A. Kaszynski 2019).

## DATA AVAILABILITY

The data used in producing this article will be shared on reasonable request to the corresponding author.

## REFERENCES

Agertz O. et al., 2007, MNRAS, 380, 963
Arjona-Gálvez Elena et al., 2025, A&A, 699, A301
Arrigoni Battaia F., Hennawi J. F., Prochaska J. X., Oñorbe J., Farina E. P., Cantalupo S., Lusso E., 2018, MNRAS, 482, 3162
Beck A. M. et al., 2015, MNRAS, 455, 2110
Begelman M. C., Cioffi D. F., 1989, ApJ, 345, L21
Benson A. J., Bower R. G., Frenk C. S., Lacey C. G., Baugh C. M., Cole S., 2003, ApJ, 599, 38
Bieri R., Dubois Y., Rosdahl J., Wagner A., Silk J., Mamon G. A., 2017, MNRAS, 464, 1854
Bigwood L., Bourne M. A., Irsic V., Amon A., Sijacki D., 2025, MNRAS, 542, 3206
Böehringer H., Voges W., Fabian A. C., Edge A. C., Neumann D. M., 1993, MNRAS, 264, L25
Booth C. M., Schaye J., 2010, MNRAS, 405, L1
Borrow J., Borrisov A., 2020, J. Open Source Softw., 5, 2430
Borrow J., Schaller M., Bower R. G., Schaye J., 2021a, MNRAS, 511, 2367
Borrow J., Schaller M., Bower R. G., Schaye J., 2021b, MNRAS, 511, 2367
Bourne M. A., Sijacki D., 2017, MNRAS, 472, 4707
Bourne M. A., Sijacki D., 2021, MNRAS, 506, 488
Bourne M. A., Yang H.-Y. K., 2023, Galaxies, 11 73
Bourne M. A., Zubovas K., Nayakshin S., 2015, MNRAS, 453, 1829
Bourne M. A., Sijacki D., Puchwein E., 2019, MNRAS, 490, 343
Bower R. G., Benson A. J., Malbon R., Helly J. C., Frenk C. S., Baugh C. M., Cole S., Lacey C. G., 2006, MNRAS, 370, 645
Braspenning J., Schaye J., Borrow J., Schaller M., 2023, MNRAS, 523, 1280
Byrne L. et al., 2024, ApJ, 973, 149
Cattaneo A., Teyssier R., 2007, MNRAS, 376, 1547
Chauhan G., Lagos C. d. P., Stevens A. R. H., Obreschkow D., Power C., Meyer M., 2020, MNRAS, 498, 44
Cole S., Lacey C. G., Baugh C. M., Frenk C. S., 2000, MNRAS, 319, 168
Costa T., Rosdahl J., Sijacki D., Haehnelt M. G., 2018, MNRAS, 479, 2079
Costa T., Pakmor R., Springel V., 2020, MNRAS, 497, 5229
Croton D. J. et al., 2006, MNRAS, 365, 11
Davé R., Anglés-Alcázar D., Narayanan D., Li Q., Rafieferantsoa M. H., Appleby S., 2019, MNRAS, 486, 2827
Davé R., Crain R. A., Stevens A. R. H., Narayanan D., Saintonge A., Catinella B., Cortese L., 2020, MNRAS, 497, 146
Devroye L., 1986, Non-Uniform Random Variate Generation. Springer, New York
Di Matteo T., Springel V., Hernquist L., 2005, Nature, 433, 604
Di Matteo T., Colberg J., Springel V., Hernquist L., Sijacki D., 2008, ApJ, 676, 33
Donnari M. et al., 2020, MNRAS, 500, 4004
Dubois Y., Devriendt J., Slyz A., Teyssier R., 2012, MNRAS, 420, 2662
Dubois Y., Gavazzi R., Peirani S., Silk J., 2013, MNRAS, 433, 3297
Dubois Y., Peirani S., Pichon C., Devriendt J., Gavazzi R., Welker C., Volonteri M., 2016, MNRAS, 463, 3948
Durier F., Dalla Vecchia C., 2012, MNRAS, 419, 465
Dutta S., Singh V., Chandra C. H. I., Wadadekar Y., Kayal A., Heywood I., 2023, ApJ, 944, 176
Eckert D., Gaspari M., Gastaldello F., Le Brun A. M. C., O'Sullivan E., 2021, Universe, 7 142
Ehlert K., Weinberger R., Pfrommer C., Pakmor R., Springel V., 2018, MNRAS, 481, 2878
Ehlert K., Weinberger R., Pfrommer C., Pakmor R., Springel V., 2022, MNRAS, 518, 4622
English W., Hardcastle M. J., Krause M. G. H., 2016, MNRAS, 461, 2025
English W., Hardcastle M. J., Krause M. G. H., 2019, MNRAS, 490, 5807
Fabian A., 2012, ARA&A, 50, 455
Fabian A. C. et al., 2000, MNRAS, 318, L65
Fabian A. C., Sanders J. S., Allen S. W., Crawford C. S., Iwasawa K., Johnstone R. M., Schmidt R. W., Taylor G. B., 2003, MNRAS, 344, L43
Fabian A. C., Sanders J. S., Taylor G. B., Allen S. W., Crawford C. S., Johnstone R. M., Iwasawa K., 2006, MNRAS, 366, 417
Falle S. A. E. G., 1991, MNRAS, 250, 581
Fanidakis N., Baugh C. M., Benson A. J., Bower R. G., Cole S., Done C., Frenk C. S., 2010, MNRAS, 410, 53
Forman W. et al., 2005, ApJ, 635, 894
Forman W. et al., 2007, ApJ, 665, 1057
Frenk C. S. et al., 1999, ApJ, 525, 554
Furlong M. et al., 2016, MNRAS, 465, 722
Gaspari M., Tombesi F., Cappi M., 2020, Nat. Astron., 4, 10
Genel S. et al., 2017, MNRAS, 474, 3976
Genel S. et al., 2019, ApJ, 871, 21
Godfrey L. E. H., Morganti R., Brienza M., 2017, MNRAS, 471, 891

Goubert P. H., Bluck A. F. L., Piotrowska J. M., Maiolino R., 2024, MNRAS, 528, 4891

Guo M., Stone J. M., Quataert E., Springel V., 2025, ApJ, 987, 202

Habouzit M. et al., 2021, MNRAS, 503, 1940

Hardcastle M. J., 2018, MNRAS, 475, 2768

Hardcastle M. J., Croston J. H., 2020, New A Rev., 88, 101539

Hardcastle M. J., Krause M. G. H., 2013, MNRAS, 430, 174

Hardcastle M. J., Krause M. G. H., 2014, MNRAS, 443, 1482

Hardee P. E., 2007, ApJ, 664, 26

Harris C. R. et al., 2020, Nature, 585, 357

Heinz S., Brüggen M., Young A., Levesque E., 2006, MNRAS, 373, L65

Henden N. A., Puchwein E., Shen S., Sijacki D., 2018, MNRAS, 479, 5385

Hitomi Collaboration et al. 2016, Nature, 535, 117

Hlavacek-Larrondo J., Li Y., Churazov E., 2022, AGN Feedback in Groups and Clusters of Galaxies. Springer Nature, Singapore, p. 1

Hopkins P. F., 2015, MNRAS, 450, 53

Hunter J. D., 2007, Comput. Sci. Eng., 9, 90

Huško F., Lacey C. G., 2023a, MNRAS, 520, 5090

Huško F., Lacey C. G., 2023b, MNRAS, 521, 4375

Huško F., Lacey C. G., Schaye J., Schaller M., Nobels F. S. J., 2022, MNRAS, 516, 3750

Huško F., Lacey C. G., Schaye J., Nobels F. S. J., Schaller M., 2023, MNRAS, 527, 5988

Ishibashi W., Fabian A. C., 2015, MNRAS, 451, 93

Jurlin N. et al., 2020, A&A, 638, A34

Kaiser C. R., Alexander P., 1997, MNRAS, 286, 215

Kaiser C. R., Best P. N., 2007, MNRAS, 381, 1548

King A., Pounds K., 2015, ARA&A, 53, 115

King A. R., Pounds K. A., 2003, MNRAS, 345, 657

Komissarov S. S., Falle S. A. E. G., 1998, MNRAS, 297, 1087

Kugel R. et al., 2023, MNRAS, 526, 6103

Laing R. A., Bridle A. H., 2013, MNRAS, 437, 3405

Li Y., Bryan G. L., 2014, ApJ, 789, 54

Mahatma V. H. et al., 2018, MNRAS, 475, 4557

Martizzi D., Quataert E., Faucher-Giguère C.-A., Fielding D., 2018, MNRAS, 483, 2465

Mattia G., Del Zanna L., Bugli M., Pavan A., Ciolfi R., Bodo G., Mignone A., 2023, A&A, 679, A49

McCarthy I. G., Schaye J., Bird S., Le Brun A. M. C., 2016, MNRAS, 465, 2936

McNamara B., Nulsen P., 2007, ARA&A, 45, 117

Menon H., Wesolowski L., Zheng G., Jetley P., Kale L., Quinn T., Governato F., 2015, Comput. Astrophys. Cosmol., 2, 1

Mignone A., Bodo G., Massaglia S., Matsakos T., Tesileanu O., Zanni C., Ferrari A., 2007, ApJS, 170, 228

Mignone A., Zanni C., Tzeferacos P., van Straalen B., Colella P., Bodo G., 2012, ApJS, 198, 7

Morsony B. J., Heinz S., Brüggen M., Ruszkowski M., 2010, MNRAS, 407, 1277

Mukherjee D., 2025, Galaxies, 13, 102

Murgia M. et al., 2011, A&A, 526, A148

Oei M. S. S. L. et al., 2024, Nature, 633, 537

Ogiya G., Biernacki P., Hahn O., Teyssier R., 2018, preprint (arXiv:1802.02177)

Pakmor R., Springel V., Bauer A., Mocz P., Munoz D. J., Ohlmann S. T., Schaal K., Zhu C., 2016, MNRAS, 455, 1134

Parma P., Murgia M., de Ruiter H. R., Fanti R., Mack K.-H., Govoni F., 2007, A&A, 470, 875

Perucho M., Martí J.-M., Quilis V., Borja-Lloret M., 2017, MNRAS, 471, L120

Pillepich A. et al., 2017, MNRAS, 473, 4077

Pope E. C. D., Babul A., Pavlovski G., Bower R. G., Dotter A., 2010, MNRAS, 406, 2023

Pounds K. A., Reeves J. N., King A. R., Page K. L., O'Brien P. T., Turner M. J. L., 2003, MNRAS, 345, 705

Power C., Read J. I., Hobbs A., 2014, MNRAS, 440, 3243

Price D. J. et al., 2018, Publ. Astron. Soc. Aust., 35, e031

Raouf M., Shabala S. S., Croton D. J., Khosroshahi H. G., Bernyk M., 2017, MNRAS, 471, 658

Read J. I., Hayfield T., 2012, MNRAS, 422, 3037

Reynolds C. S., McKernan B., Fabian A. C., Stone J. M., Vernaleo J. C., 2005, MNRAS, 357, 242

Rosswog S., Price D., 2007, MNRAS, 379, 915

Rupke D. S. N., Veilleux S., 2011, ApJ, 729, L27

Saitoh T. R., Makino J., 2009, ApJ, 697, L99

Saitoh T. R., Makino J., 2013, ApJ, 768, 44

Sandnes T., Eke V., Kegerreis J., Massey R., Ruiz-Bonilla S., Schaller M., Teodoro L., 2025, J. Comput. Phys., 532, 113907

Sazonov S. Y., Ostriker J. P., Ciotti L., Sunyaev R. A., 2005, MNRAS, 358, 168

Schaller M. et al., 2024, MNRAS, 530, 2378

Schaye J. et al., 2014, MNRAS, 446, 521

Scheuer P. A. G., 1974, MNRAS, 166, 513

Sembolini F. et al., 2016, MNRAS, 457, 4063

Shabala S. S., Jurlin N., Morganti R., Brienza M., Hardcastle M. J., Godfrey L. E. H., Krause M. G. H., Turner R. J., 2020, MNRAS, 496, 1706

Sijacki D., Springel V., Di Matteo T., Hernquist L., 2007, MNRAS, 380, 877

Sijacki D., Vogelsberger M., Kereš D., Springel V., Hernquist L., 2012, MNRAS, 424, 2999

Sijacki D., Vogelsberger M., Genel S., Springel V., Torrey P., Snyder G. F., Nelson D., Hernquist L., 2015, MNRAS, 452, 575

Silk J., Rees M. J., 1998, A&A, 331, L1

Spilker J. S., Champagne J. B., Fan X., Fujimoto S., van der Werf P. P., Yang J., Yue M., 2025, ApJ, 982, 72

Springel V., 2005, MNRAS, 364, 1105

Springel V., 2010, MNRAS, 401, 791

Springel V., Pakmor R., Zier O., Reinecke M., 2021, MNRAS, 506, 2871

Stewart G. S. C., Shabala S. S., Turner R. J., Yates-Jones P. M., Krause M. G. H., Wong O. I., Power C., Hardcastle M. J., 2025, Publ. Astron. Soc. Aust., 42, e152

Su K.-Y. et al., 2021, MNRAS, 507, 175

Sullivan C. B., Kaszynski A. A., 2019, J. Open Source Softw., 4, 1450

Talbot R. Y., Bourne M. A., Sijacki D., 2021, MNRAS, 504, 3619

Talbot R. Y., Sijacki D., Bourne M. A., 2022, MNRAS, 514, 4535

Talbot R. Y., Sijacki D., Bourne M. A., 2024, MNRAS, 528, 5432

Teyssier R., 2002, A&A, 385, 337

Thomas N., Davé R., Jarvis M. J., Anglés-Alcázar D., 2021, MNRAS, 503, 3492

Turner R. J., Shabala S. S., 2023, Galaxies, 11 87

Turner R. J., Yates-Jones P. M., Shabala S. S., Quici B., Stewart G. S. C., 2023, MNRAS, 518, 945

van Rossum G., 1995, Python Reference Manual. Centrum voor Wiskunde en Informatica, Amsterdam

van Weeren R. J. et al., 2024, A&A, 692, A12

Vernaleo J. C., Reynolds C. S., 2006, ApJ, 645, 83

Virtanen P. et al., 2020, Nat. Methods, 17, 261

Vogelsberger M., Marinacci F., Torrey P., Puchwein E., 2020, Nat. Rev. Phys., 2, 42

Voit G. M., Kay S. T., Bryan G. L., 2005, MNRAS, 364, 909

Volonteri M., Dubois Y., Pichon C., Devriendt J., 2016, MNRAS, 460, 2979

Wadsley J. W., Veeravalli G., Couchman H. M. P., 2008, MNRAS, 387, 427

Wadsley J. W., Keller B. W., Quinn T. R., 2017, MNRAS, 471, 2357

Weinberger R. et al., 2016, MNRAS, 465, 3291

Weinberger R., Ehlert K., Pfrommer C., Pakmor R., Springel V., 2017, MNRAS, 470, 4530

Weinberger R., Springel V., Pakmor R., 2020, ApJS, 248, 32

Weinberger R. et al., 2023, MNRAS, 523, 1104

Wright R. J., Somerville R. S., Lagos C. d. P., Schaller M., Davé R., Anglés-Alcázar D., Genel S., 2024, MNRAS, 532, 3417

Yang H. Y. K., Reynolds C. S., 2016, ApJ, 818, 181

Yates-Jones P. M., Shabala S. S., Krause M. G. H., 2021, MNRAS, 508, 5239

Yates-Jones P. M., Shabala S. S., Power C., Krause M. G. H., Hardcastle M. J., Mohd Noh Velastín E. A. N., Stewart G. S. C., 2023, Publ. Astron. Soc. Aust., 40, e014

Yates P. M., Shabala S. S., Krause M. G. H., 2018, MNRAS, 480, 5286

## APPENDIX A: DISTRIBUTION OF VELOCITIES

The coupling between hydrodynamical code and subgrid jet model is an important aspect of jet modelling and should be treated carefully. In our simulations, we used different injection velocities and injection schemes which led to distinct jet behaviours. To examine these differences in detail, we analyse the velocity distributions of individual elements in our simulations and how these distributions vary between runs.

In Fig. A1, we plot the mass-weighted velocity distributions of resolution elements for the different runs with the three codes. We include runs with the different jet injection velocities in separate rows and the two injection schemes in the two columns. The left column shows the neighbour-number scheme (see Section 3.2) and the right column shows the neighbour-mass scheme (see Section 4.4). At the lowest injection velocity ($v_\mathrm{j} = 2 \times 10^4\,\mathrm{km\,s^{-1}}$), the distributions for the three codes show reasonable agreement for both injection schemes. For both the fiducial ($v_\mathrm{j} = 4 \times 10^4\,\mathrm{km\,s^{-1}}$) and the highest ($v_\mathrm{j} = 8 \times 10^4\,\mathrm{km\,s^{-1}}$) injection velocities, PLUTO cells achieve higher velocities than the other two codes. This effect is significantly reduced in the case of the mass-based injection scheme. As discussed in the main text, in our fiducial neighbour-number scheme, the neighbour mass in PLUTO will be much lower than in SWIFT and AREPO, as the latter codes employ (roughly) fixed mass resolution. As a result, for fixed kinetic energy injections, PLUTO cells will achieve higher velocities than SWIFT particles and AREPO cells. In the fixed neighbour-mass scheme, however, this effect will naturally be corrected.

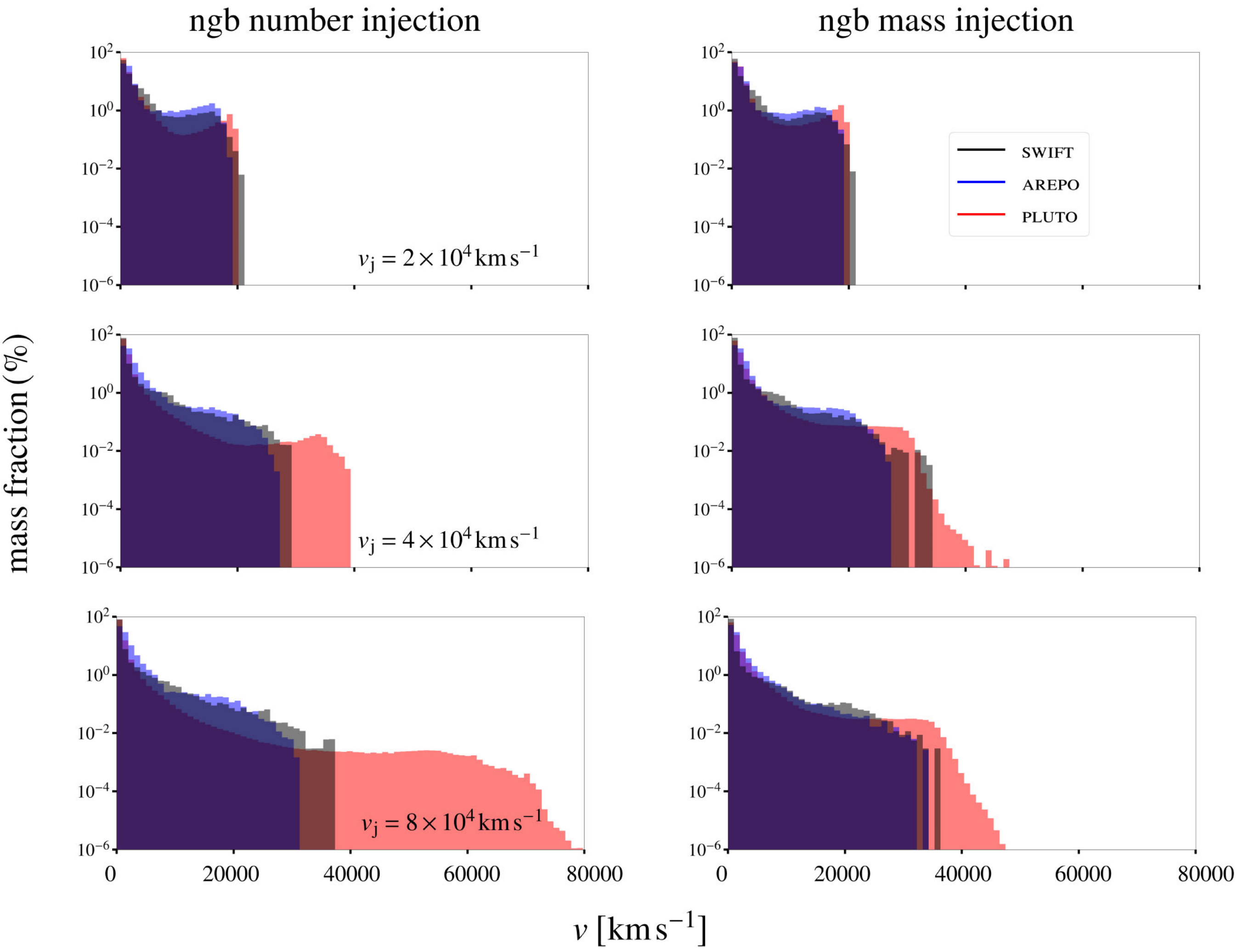

**Figure A1.** Mass-weighted velocity distributions of resolution elements for the different runs with the three codes. Rows correspond to the different jet injection velocities, while columns correspond to the two injection schemes: fixed neighbour number and fixed neighbour mass. At higher injection velocities, PLUTO cells achieve higher velocities than the other codes but significantly less so for the mass-based injection scheme.

This paper has been typeset from a TEX/LATEX file prepared by the author.