Numerical simulations of the galactic-scale interstellar medium

Simon Glover

Physical ingredients

Hydrodynamics

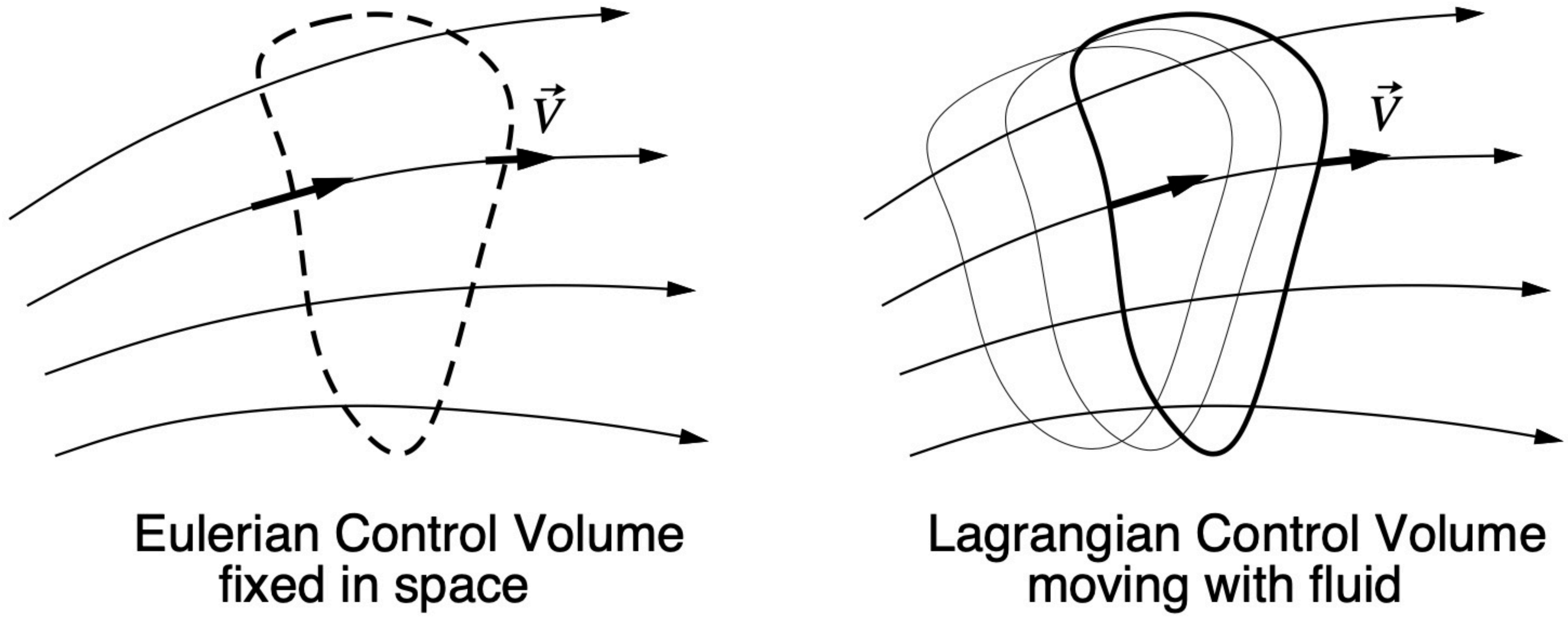
- The ISM is composed almost entirely of gas
  - Solid matter (dust) accounts for ~1% of total mass budget at solar Z, much less in metal-poor galaxies
  - Cosmic rays account for negligible mass fraction
- Any numerical model of the time-varying ISM needs to account for the flow of gas, i.e. **hydrodynamics**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0,$$

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{v} \boldsymbol{v}^{\mathrm{T}} + \boldsymbol{I} p_{\mathrm{tot}} \right) = -\rho \nabla \Phi,$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (\boldsymbol{v}\,(E + p_{\mathrm{tot}})) = -\rho\,(\boldsymbol{v} \cdot \nabla \Phi) + \mathcal{H} - \Lambda,$$

- Basic equations for mass, momentum, energy of gas
- First decision: how do we discretise and solve?



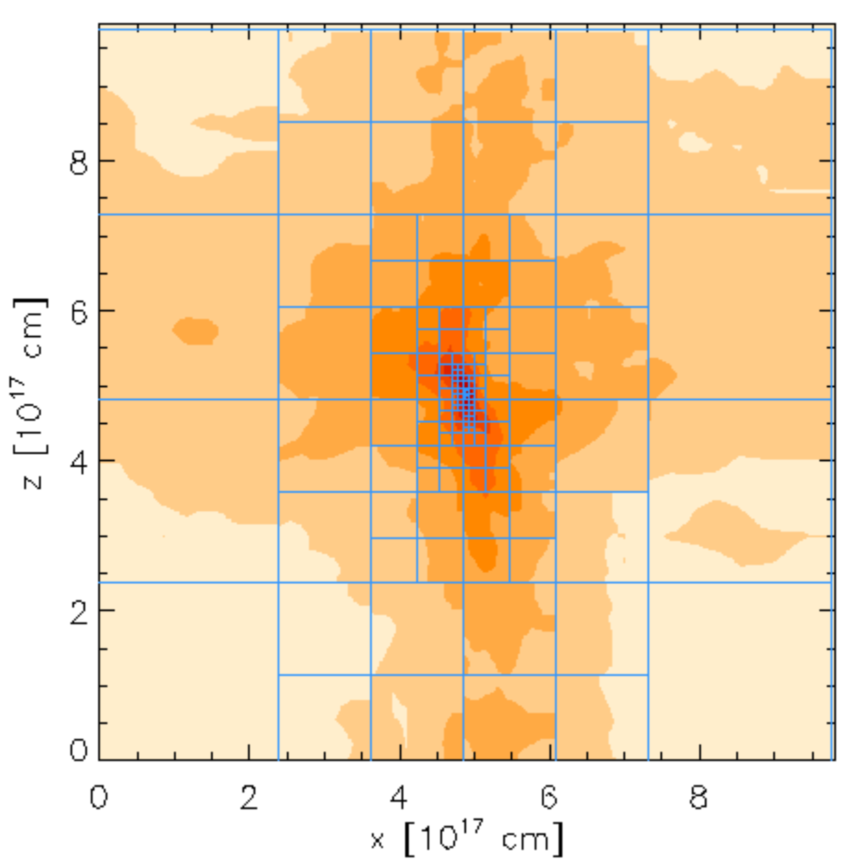
- Two different ways to formulate fluid equations:
  - **Eulerian**: control volume fixed in space
  - **Lagrangian**: control volume moving with fluid

- Can carry this distinction over to our discretisation
- Eulerian:
  - Spatial grid fixed in space, gas flows in and out of grid cells
  - Most commonly Cartesian, but cylindrical, spherical grids also relatively common (e.g. PLUTO)
- Lagrangian:
  - Fluid represented by set of Lagrangian fluid elements (grid cells or “particles”)
  - Elements move with the flow, changing position and size/shape
  - Each element conserves total mass

- Fixed Eulerian grid: simple, fast but has limited dynamical range

- Structures in ISM cover broad range of scales (>> 100 pc for spiral arms, bar, etc., << 1 pc for prestellar core)

- Get higher dynamical range in Eulerian approach using **adaptive mesh refinement** (AMR)

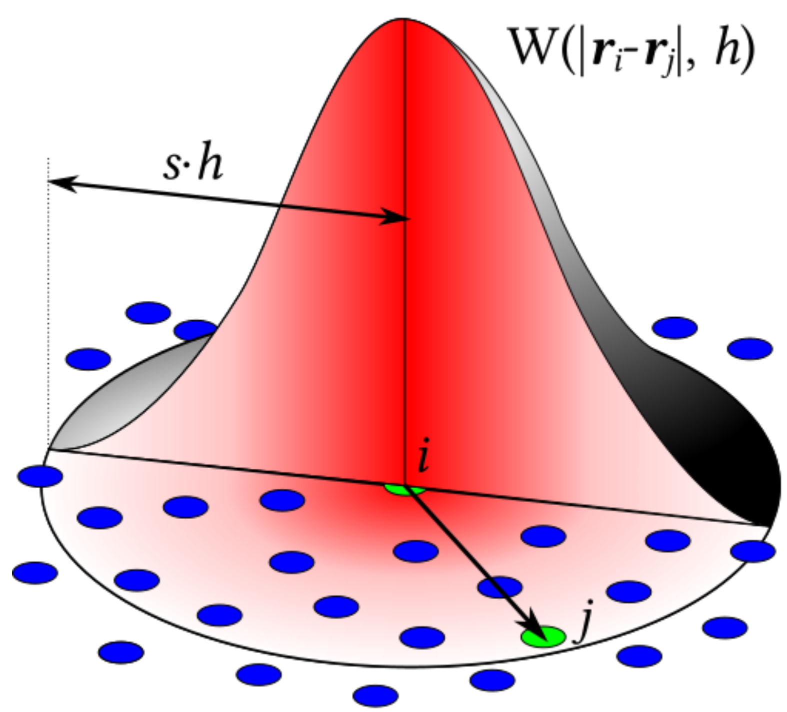


- Advantages of AMR approach:
  - Simple grid structure
  - Straightforward to implement new physics
  - Good resolution even at low densities
- Disadvantages of AMR approach:
  - Often more costly at high resolution than Lagrangian approach
  - Less accurate for representing large-scale bulk flows
- Examples: FLASH, RAMSES

- Main Lagrangian method used to simulate the ISM: **smoothed particle hydrodynamics**

- Basic idea: model fluid with set of particles with fixed masses that move with the flow

- Physical quantities can be derived by summing over particle properties with suitable kernel function

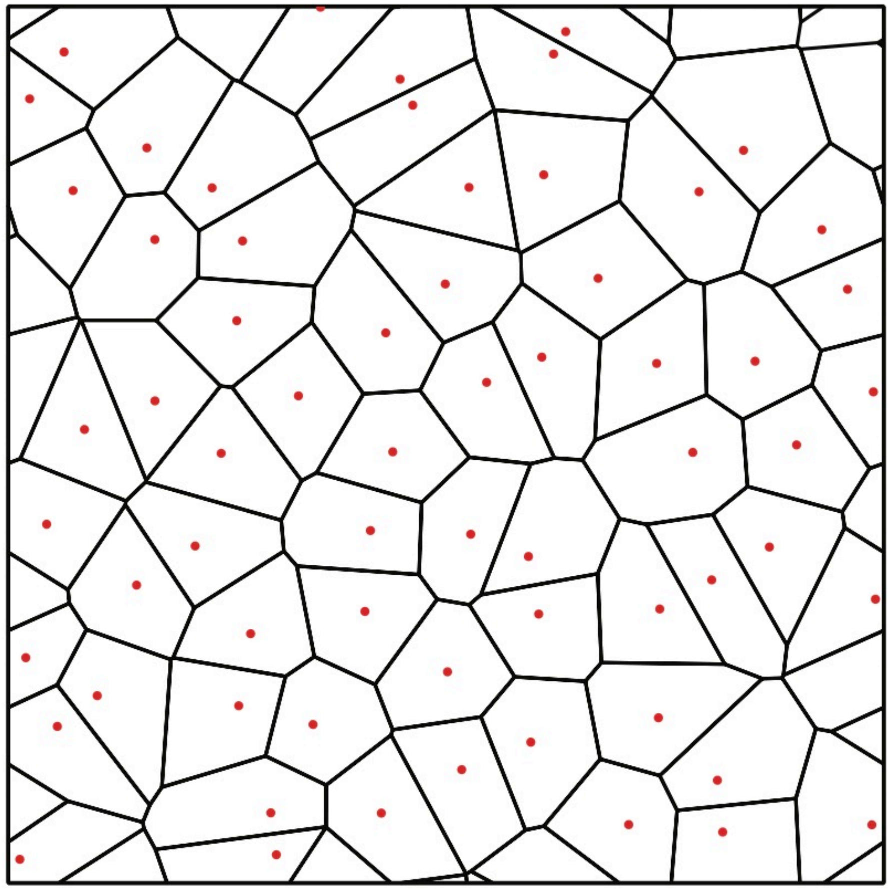


Credit: Wikipedia

- Advantages of SPH:
    - In most basic form, very simple to implement
    - Mesh free — hence no grid artifacts
    - Exactly conserves linear and angular momentum
- Disadvantages of SPH:
    - Limited dynamical range — particle splitting not reliable if particles of different mass mix
    - Requires artificial viscosity (and sometime artificial thermal conduction) for stability, correctness
    - Behaviour can depend on choice of smoothing kernel — e.g. tensile instability
- Examples: Phantom, GADGET

Mixed methods

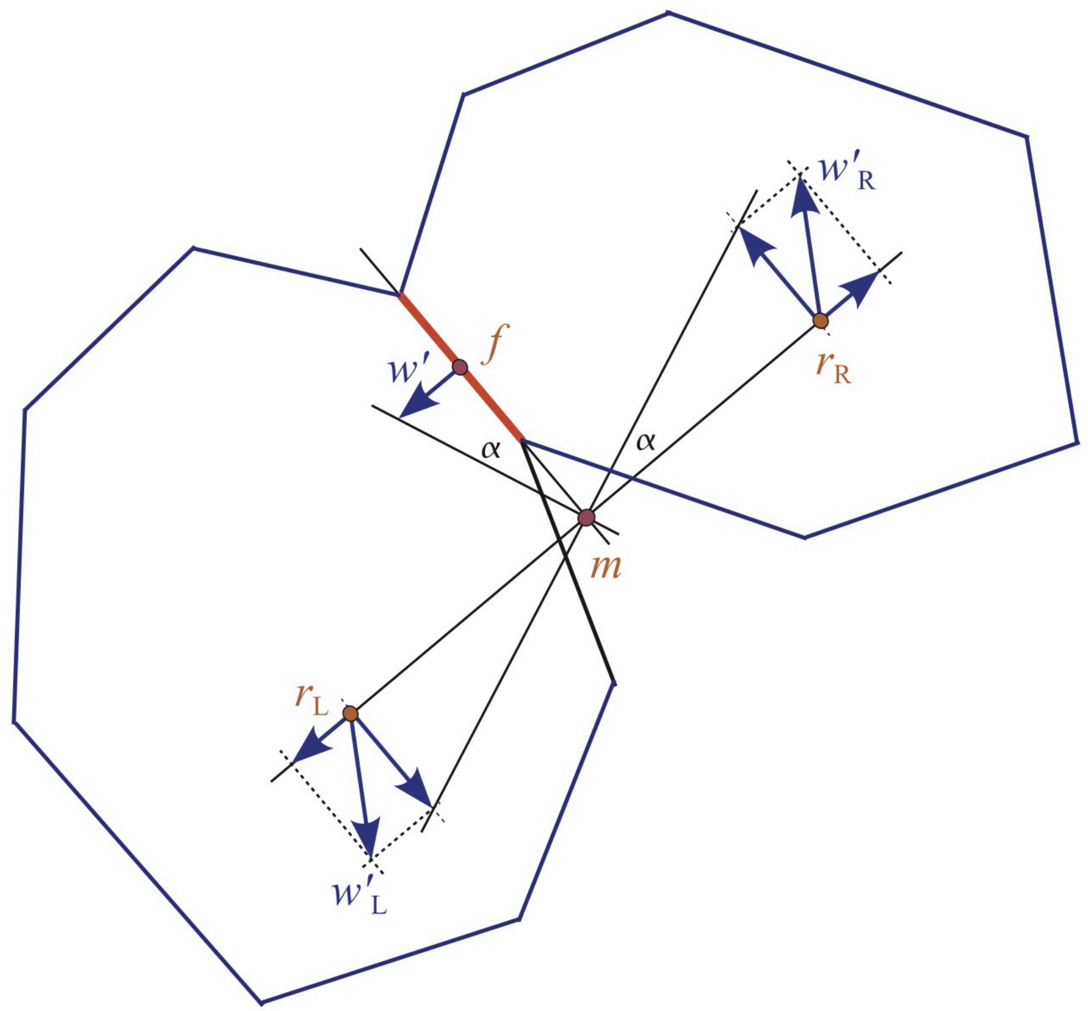
- Would like to combine strengths of Eulerian (AMR), Lagrangian approaches in a single code
- Two main versions of this approach in common usage for ISM simulations:
  - Moving unstructured mesh (e.g. AREPO)
  - Mesh-free quasi-Lagrangian approach (e.g. GIZMO)



Springel (2010)

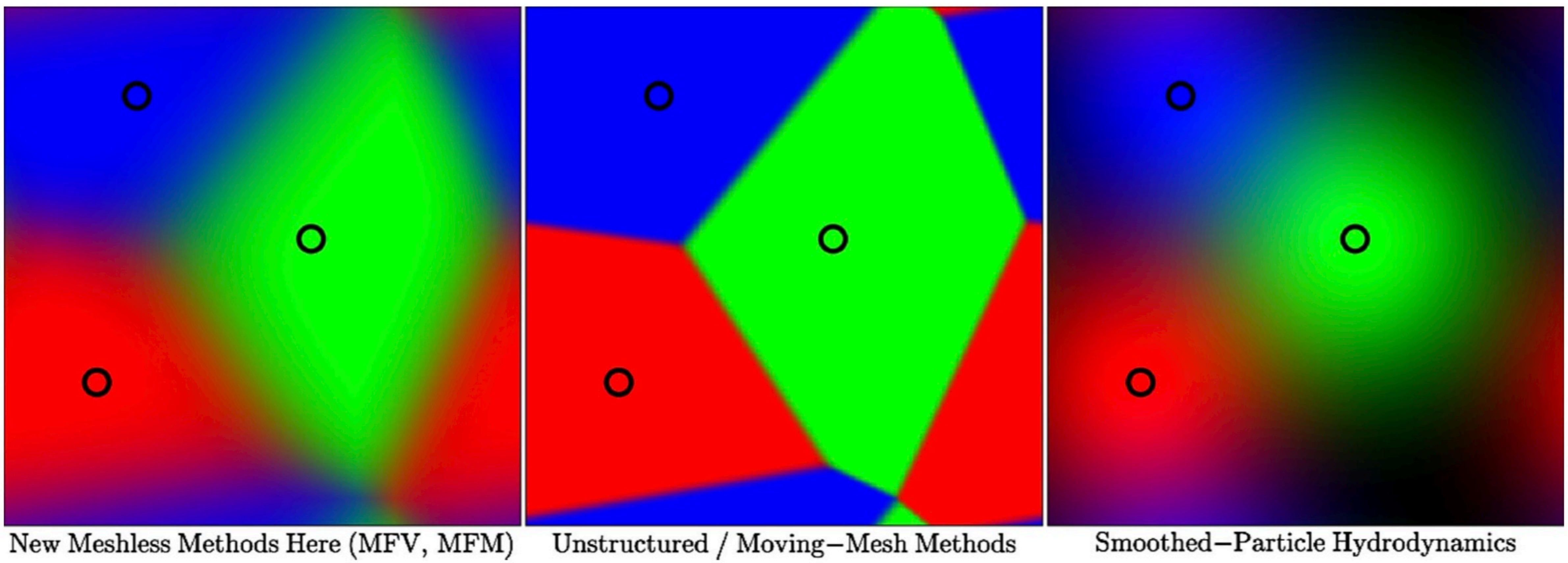
- AREPO:
  - Define set of sample points that move with the flow
  - Mesh is the Voronoi tesselation of these points
  - Flow variables within each cell represent mass-weighted average over volume of cell

- Avoid mesh distortion by applying corrective velocities to mesh generating points (“regularisation”)
- Not fully Lagrangian — need to compute fluxes between cells



- Advantages of AREPO:
    - Galilean invariant (as with SPH); good choice for problems with high bulk velocities
    - Highly adaptive — simply add additional mesh generating points (or remove to de-refine)
    - Many different physics modules
- Disadvantages of AREPO:
    - Highly complex — difficult to add new physics
    - Public version has very limited set of modules — most interesting stuff for ISM is in private version

- Mesh-less finite mass (MFM):
  - Define set of sampling points
  - For each differential spatial volume, partition amongst closest particles using weighting function
  - In case where weighting function assigns all volume to closest point, reduces to Voronoi mesh
  - More generally, get Voronoi mesh with fuzzy borders



Hopkins (2015)

- Particle velocities set to match fluid velocities
- Solve for fluxes between volumes assigned to different particles
- Can design flux calculation so that mass fluxes are zero (i.e. each particle conserves mass) — this yields the MFM method
- NOTE: can have non-zero fluxes of other quantities (momentum, energy etc.) even when net mass flux is zero

- Advantages of MFM:
  - Galilean invariance, highly adaptive
  - Lower memory cost than AREPO
  - Many physics modules
- Disadvantages:
  - Complexity
  - Limited set of physics modules in public version
- Examples: GIZMO

Magnetohydrodynamics

- ISM is magnetised, so should actually solve MHD equations rather than hydrodynamics equations

- Additional terms simple to include

- Main issue: what to do about div(B)?

- Physically, div(B) = 0 everywhere and at all times; but numerically, not always guaranteed to be the case

Dealing with div(B)

- If possible, use an algorithm that enforces div(B) = 0 by construction — e.g. constrained transport

- However, not always available/practical — e.g. in Arepo, constrained transport cannot be used together with hierarchical timesteps

- Next best option: divergence cleaning — propagate divergence errors to domain boundaries and damp

- Widely used, but need to ensure timestep small enough to remove div(B) faster than it grows

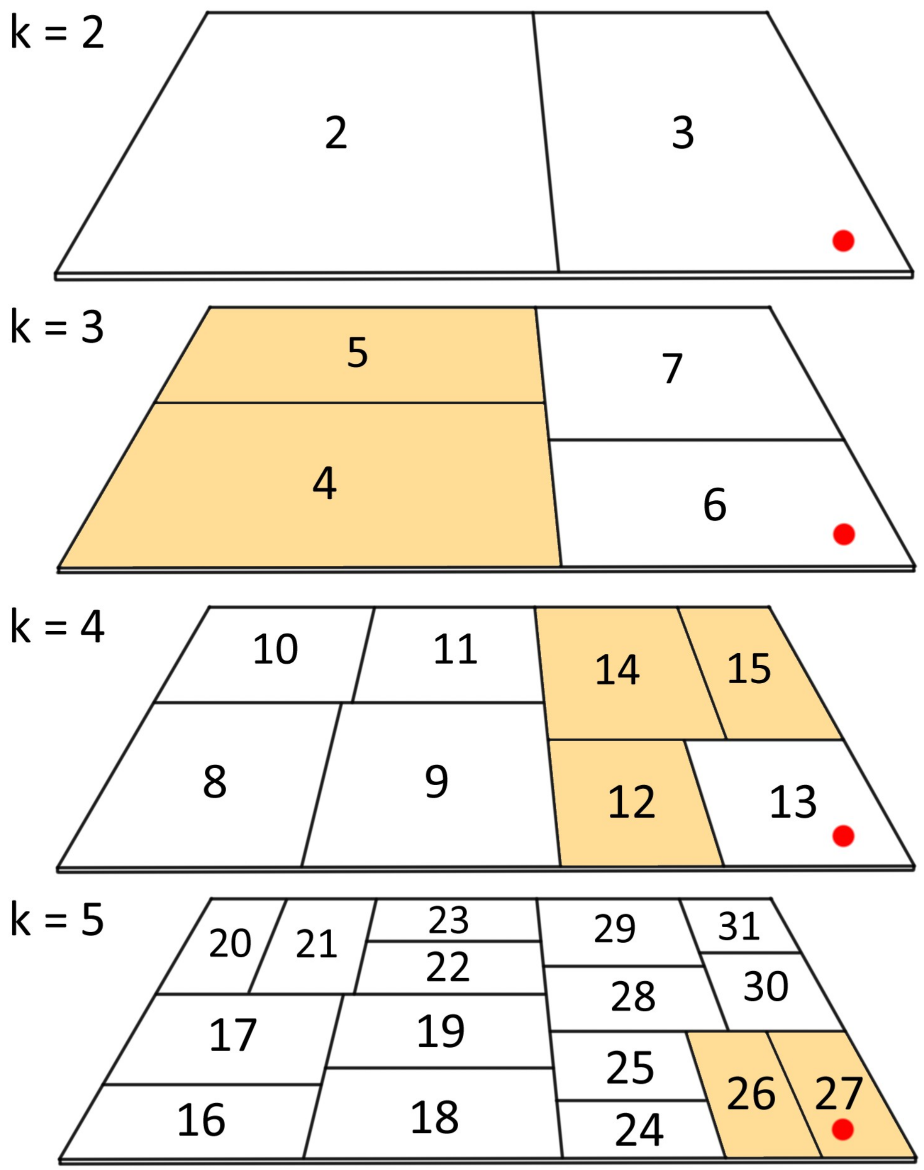
Other B field issues

- Non-ideal effects
  - Generally unimportant on > 1 pc scales, but could become significant in extreme zoom-ins
- Initial conditions
  - Start with weak seed field and grow via dynamo or start with prescribed field strength and geometry
- What do you do with B when stars form?
  - Removing gas but not magnetic energy can lead to unphysical local decrease in plasma beta

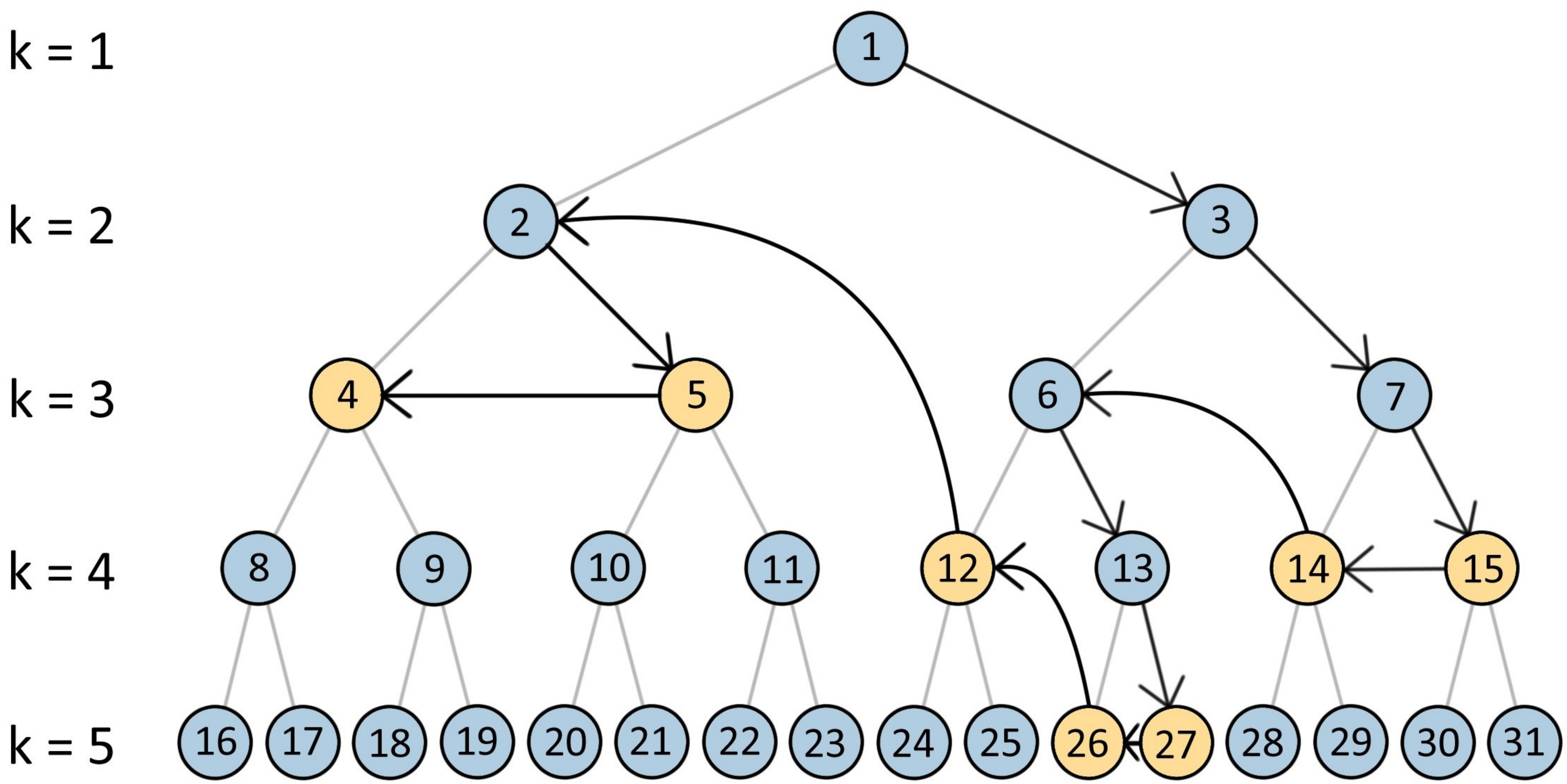
Gravity

- Computing gravitational forces by direct summation highly inefficient — cost scales as N^2
- Most ISM simulations use some form of tree code
- Basic idea:
  - group together distant particles
  - compute force assuming all particles located at centre of mass of group
  - error depends on size of group vs. distance to group; can be controlled by making this ratio small

- Simple and efficient way to put this into practice is by organising particles into hierarchical tree structure



Lau et al (2024)



Lau et al (2024)

- Different types of tree exist (e.g. binary tree, oct tree), but all versions share same basic idea

- Cost of computing gravity with tree scales as N log N

- Massive speed-up compared to direct summation!

- Disadvantages: complexity, (minor) loss of accuracy

Gravitational softening

- To avoid (unphysically) high gravitational accelerations on small scales, common to **soften** gravity

- E.g. Plummer softening:

$$F_{\mathrm{soft}} = \frac{G m_1 m_2}{(|\vec{r}|^2 + \epsilon^2)^{3/2}} \vec{r}$$

- Softening length can be explicitly set or related to local size of grid cells (or SPH smoothing length)

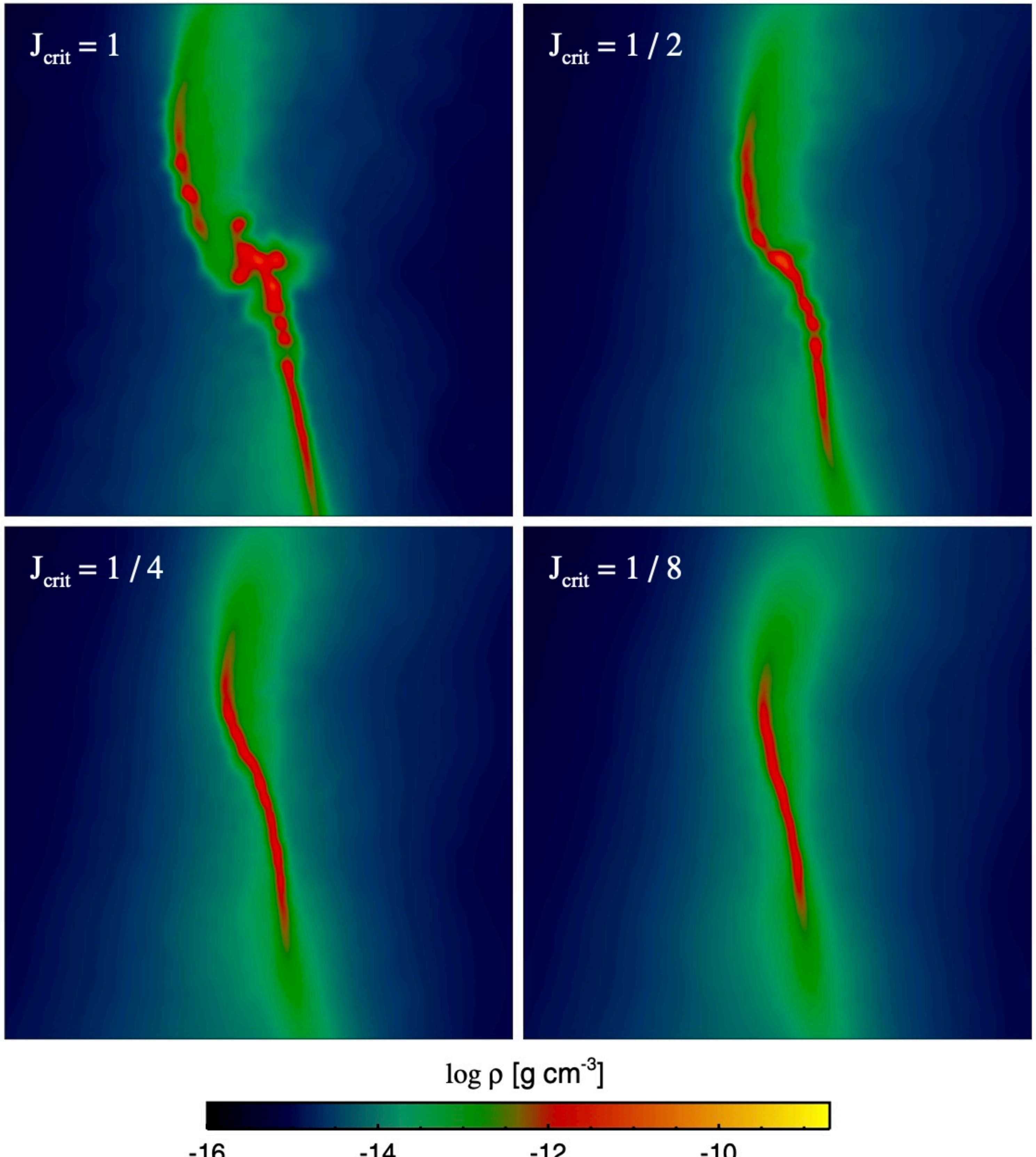
- Fixed softening length simple, but not a good choice when dealing with self-gravitating gas

- For gas self-gravity: softening, smoothing of pressure gradient due to grid resolution must match

- Otherwise, can get artificial fragmentation (if pressure smoothed more than gravity), artificial suppression of fragmentation (if gravity smoothed more than pressure)

Truelove criterion

- Gas self-gravity introduces another issue: to avoid artificial fragmentation, must resolve **Jeans length**
- Early analysis by Truelove et al. found that at least 4 resolution elements per Jeans length needed — this has become known as the Truelove criterion
- Simulations often take a larger number — 8 is common, 16 less so
- More = better, but more also = more costly…



Greif et al. (2011)

Gravity from stars, dark matter etc.

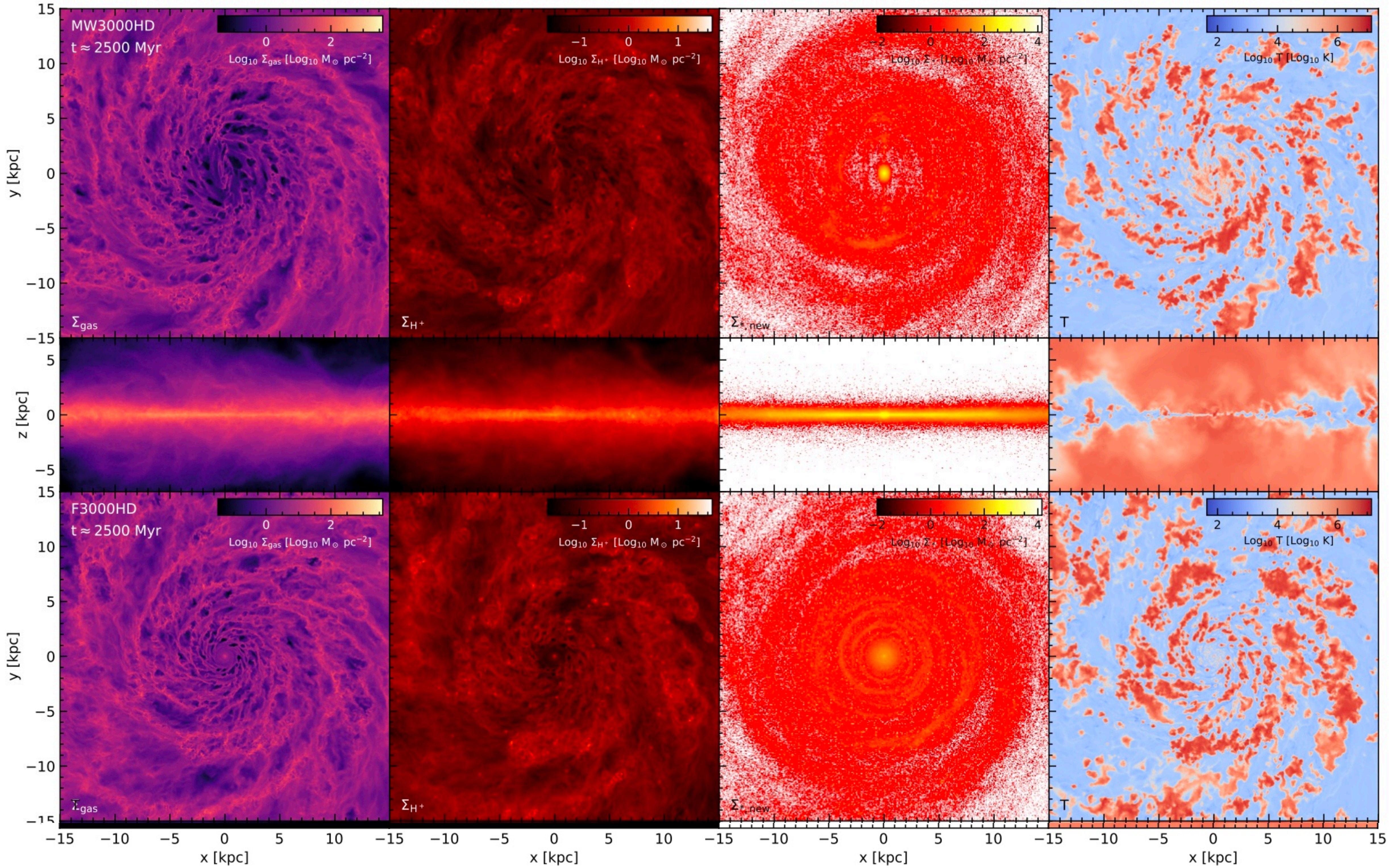
- In most galaxies, gas is small fraction of total mass
- Also need to account for gravitational forces from other components: stars, dark matter, in some cases also the central supermassive black hole
- Young stars often tracked explicitly — see later
- What about old stars, dark matter?
- Two main options: explicitly follow dynamical evolution ("live halo") or prescribe fixed spatial distribution

- Fixing spatial distribution of old stars, dark matter is highly computationally efficient

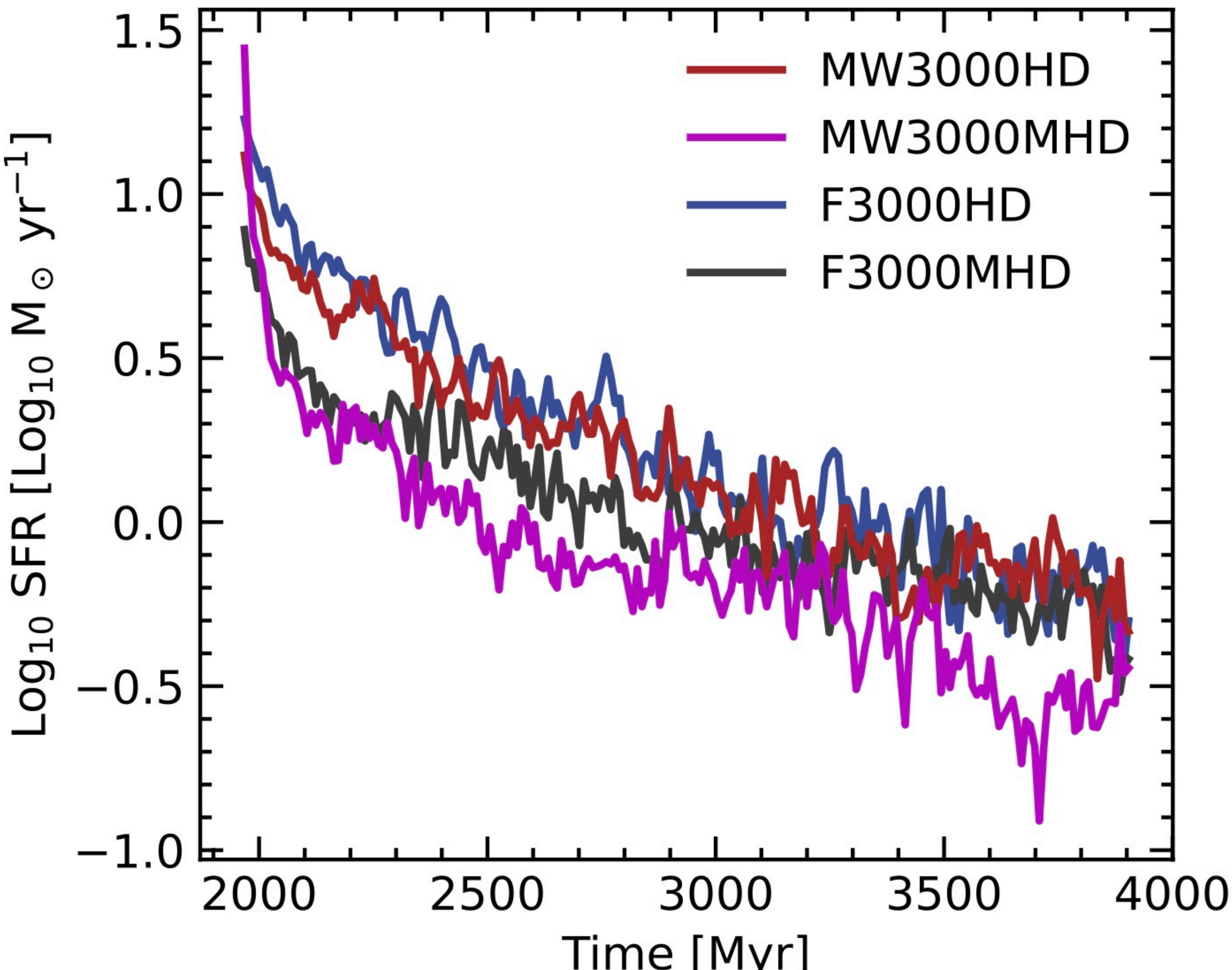
- Also best choice for simulations designed to represent specific galaxies (e.g. the Milky Way)

- Models can be very simple (bulge, disk, DM halo) or much more complicated (e.g. including nuclear stellar cluster, bar, spiral arms etc.)

- Substantial effect on gas morphology, minor effect on other galactic properties (e.g. star formation rate)



Göller et al. (2025)



Göller et al. (2025)

Star formation

- Gravitationally collapsing gas increases its density, decreases its Jeans length (L_J)
- What happens once we can no longer resolve L_J?
- One option: locally improve resolution so that L_J remains well resolved — **Jeans refinement**
- This helps improve dynamical range, but eventually becomes too costly to continue. What then?
- Two options: stop the simulation, or allow stars to form

- Wide variety of different star formation algorithms
- Simplest option: density threshold
  - All gas with density above threshold converted to stars on user-prescribed timescale
  - Can also be combined with constraints on e.g. temperature, H_2 content
  - Main disadvantage: no guarantee that gas is actually gravitationally bound!

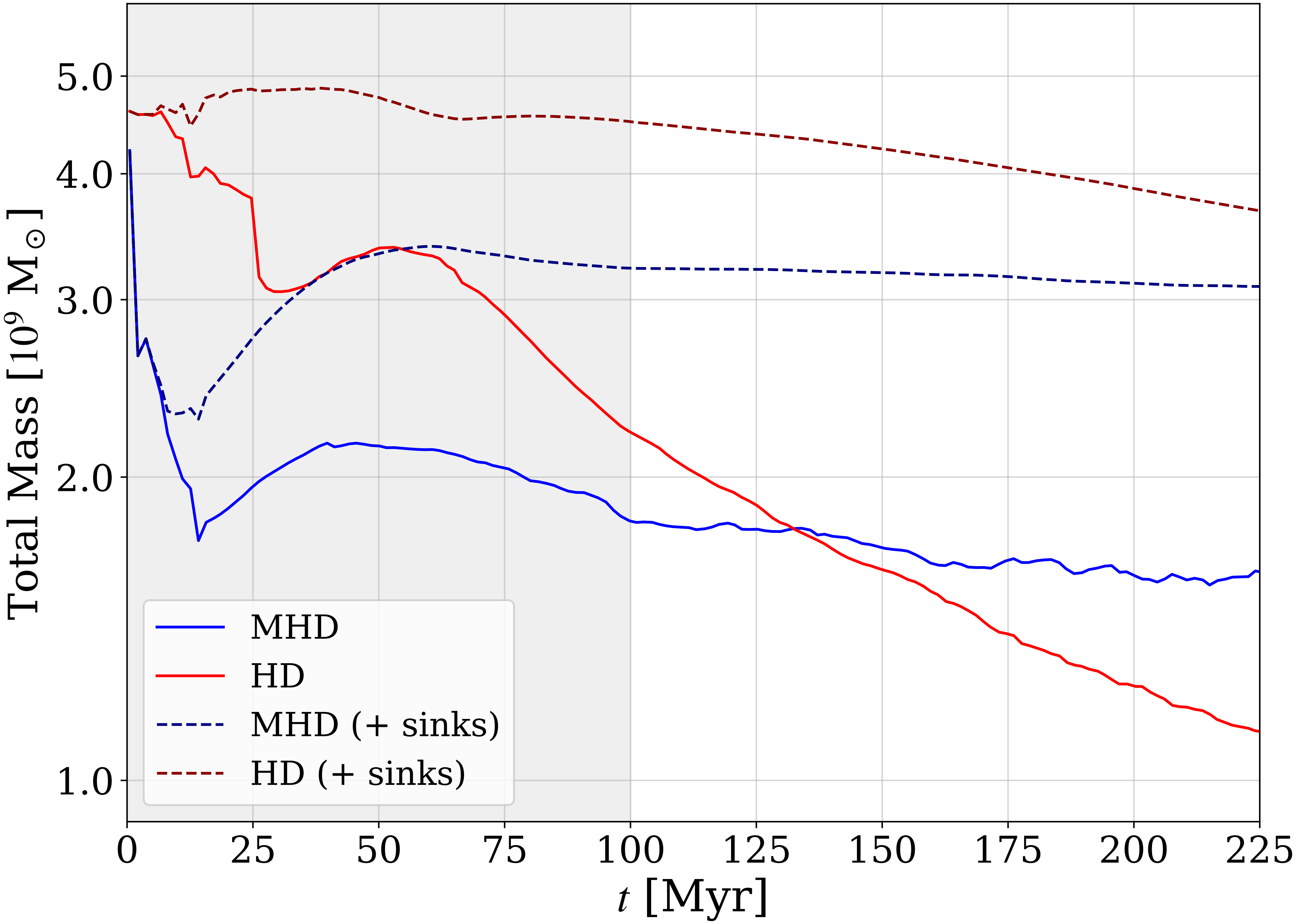
- Generally safer to add additional checks:
  - Gravitationally bound (i.e. total energy < 0)
  - Collapsing (negative velocity divergence)
  - Accelerating inwards (negative div(a))
- Once you decide gas should form stars, still many options:
  - Convert instantly to collisionless “star” particle
  - Convert to star particle on set timescale (e.g. local free-fall time) with prescribed efficiency
  - Convert to sink particle

Star particles

- Collisionless particles — interaction only via gravity
- Mass fixed at formation
- Depending on mass, may either represent individual stars or stellar clusters
- Need to take care that stars formed properly sample assumed IMF

Sink particles

- Collisionless particles
- Mass **not** fixed — can accrete gas from surroundings
- Again, can represent individual stars or clusters
- Take care with IMF sampling
- If SF efficiency not 100%, dense gas can end up trapped in sinks — must be accounted for when analysing gas properties



Bogue et al., in prep.

- Advantages of star particles:
    - Lightweight — mass, age, position, velocity
    - Little communication (other than gravity tree)

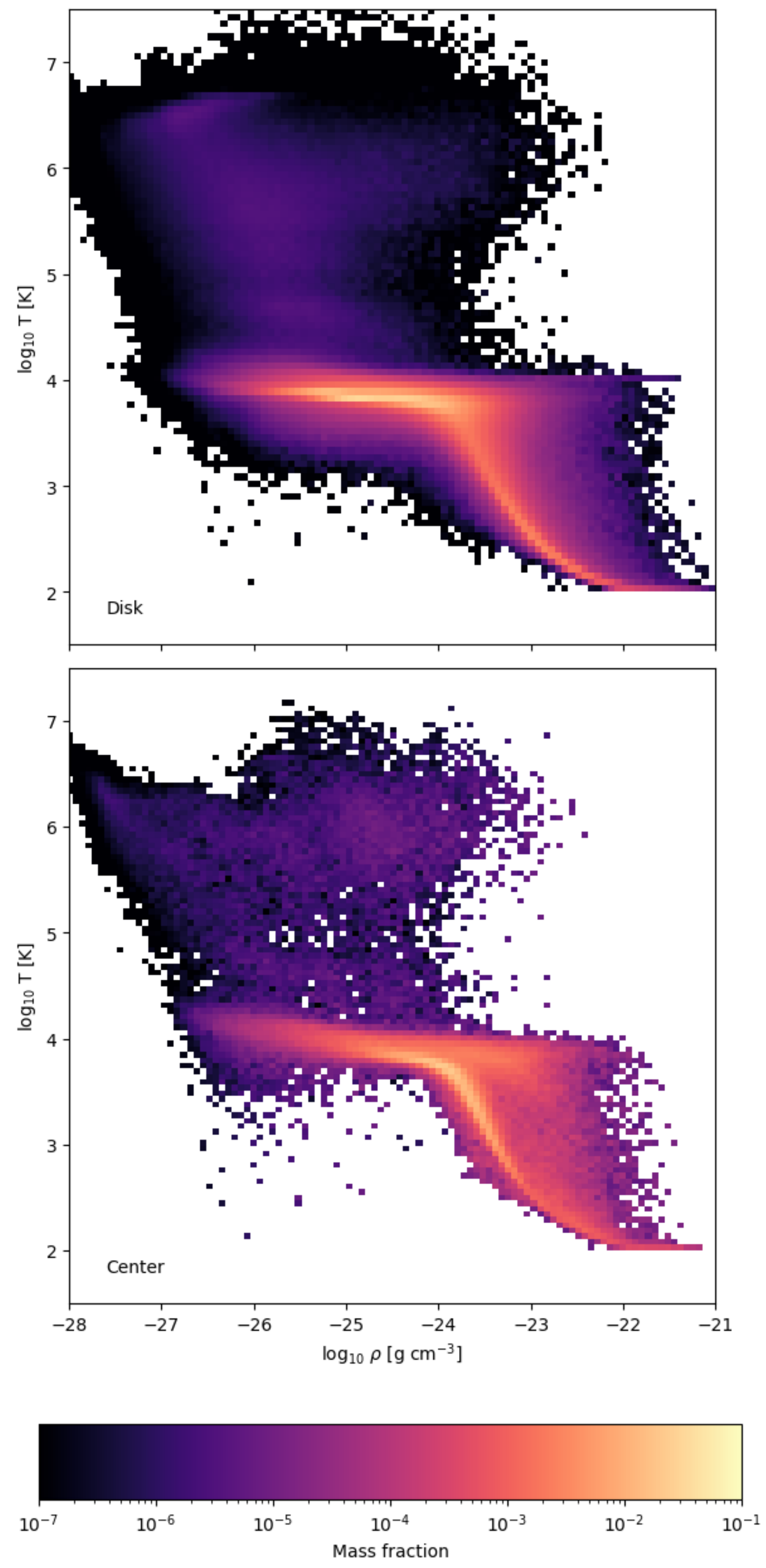
- Disadvantages of star particles:
    - Number of particles can become very large
    - Not always effective at preventing gas from collapsing beyond resolution limit

- Advantages of sink particles:
    - Better choice for small scale simulations, since allow actual mass, angular momentum etc. accumulated during pre-MS phase to be measured

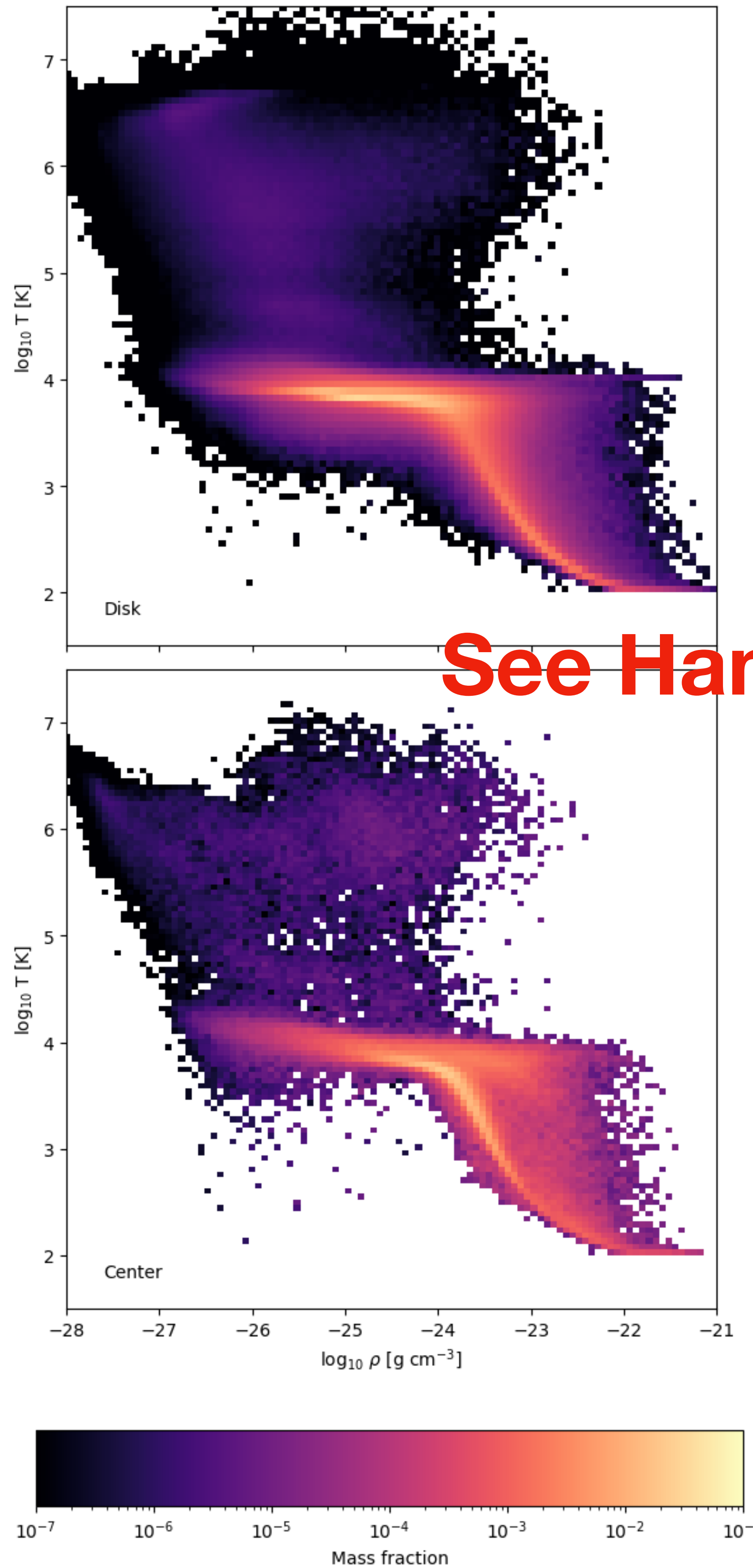
- Disadvantages of sink particles:
    - Gas trapping in sinks
    - Higher level of communication needed, so scales very poorly to large numbers of sinks

Cooling

- ISM contains gas with a huge range of temperatures
- At some densities, gas close to thermal equilibrium
- At other densities, significant scatter in T at given density — influence of non-equilibrium effects
- Can't assume that thermal equilibrium is always valid — need to actually solve for evolution of temperature
- Note: this is for gas; for dust, can almost always assume thermal equilibrium

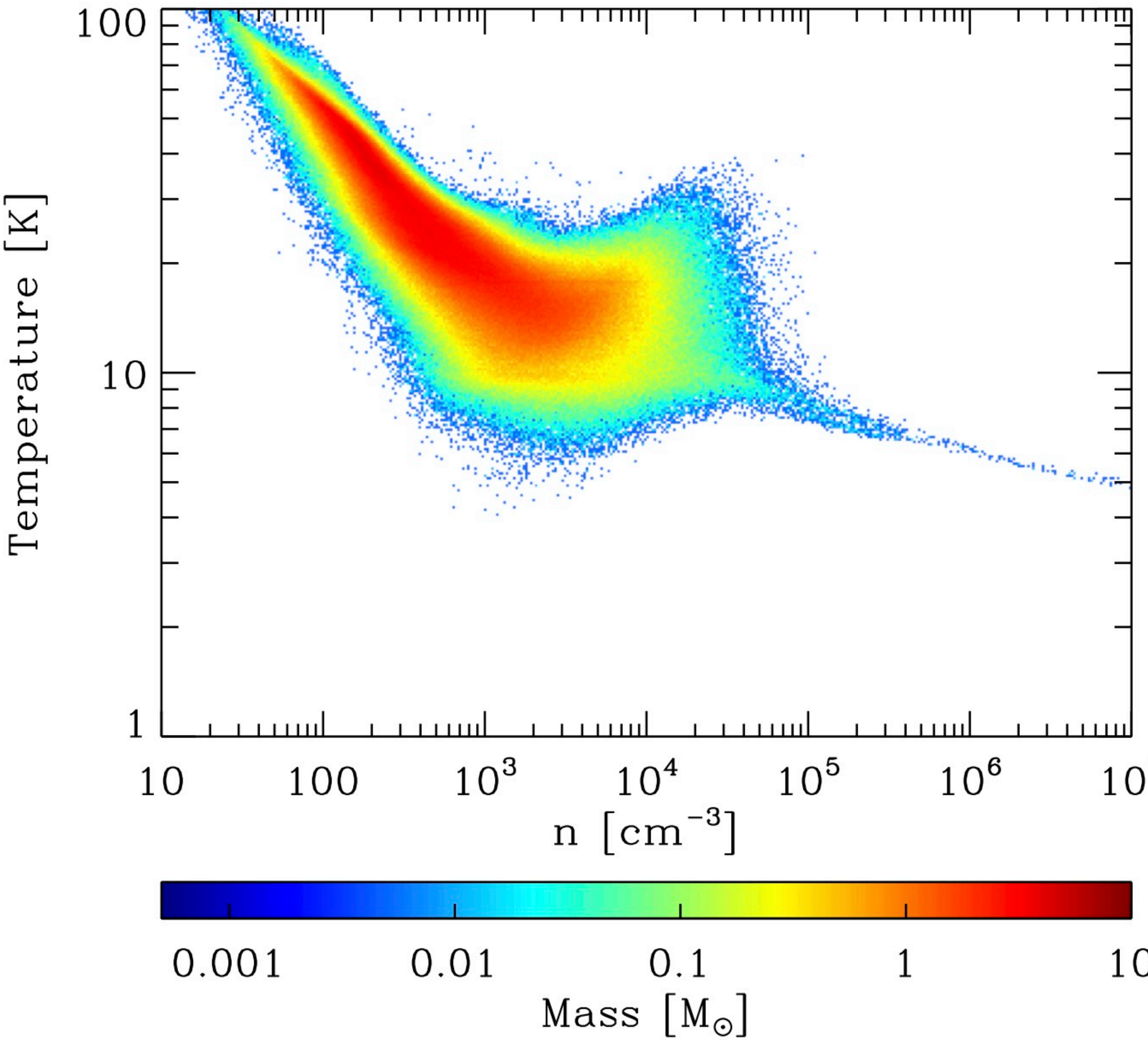


Based on data from Kjellgren et al. (2025)



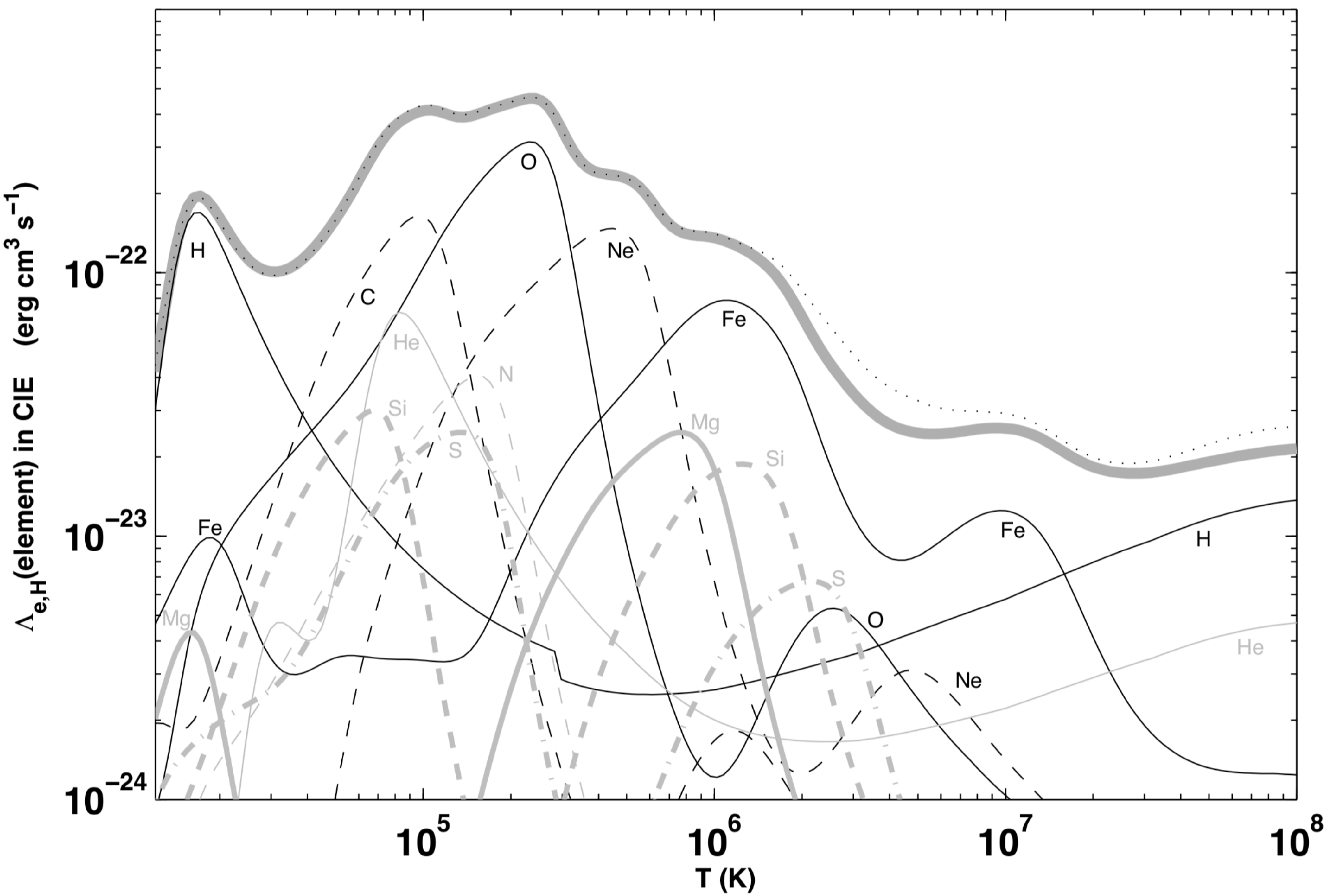
See Hands On project #9!

Based on data from
Kjellgren et al. (2025)



Glover et al. (2015)

- At high temperatures (T > 10000 K), cooling dominated by permitted transitions of H, He, metals, plus free-free
- Accurately solving for cooling rate in this regime involves tracking ionisation states of many metals
  - Complicated chemical calculation, very costly
- Common to approximate by assuming collisional ionisation equilibrium (CIE) — then cooling becomes a simple function of density, temperature

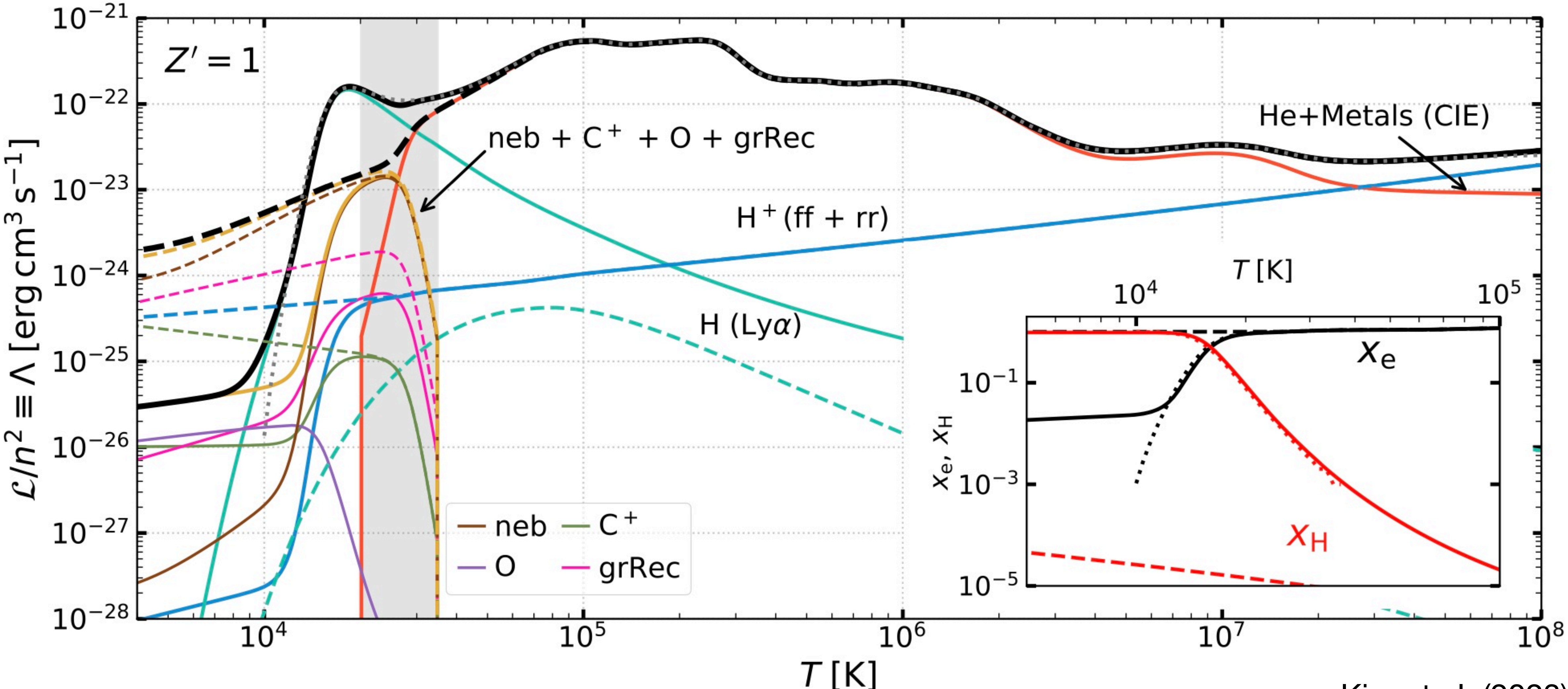


Gnat & Ferland (2012)

- Problems with this approach:
  - Non-equilibrium effects become very important close to 10000 K — cooling time shorter than chemical timescale
  - Doesn’t do a good job of modelling cooling in e.g. HII regions, which end up with T far too low

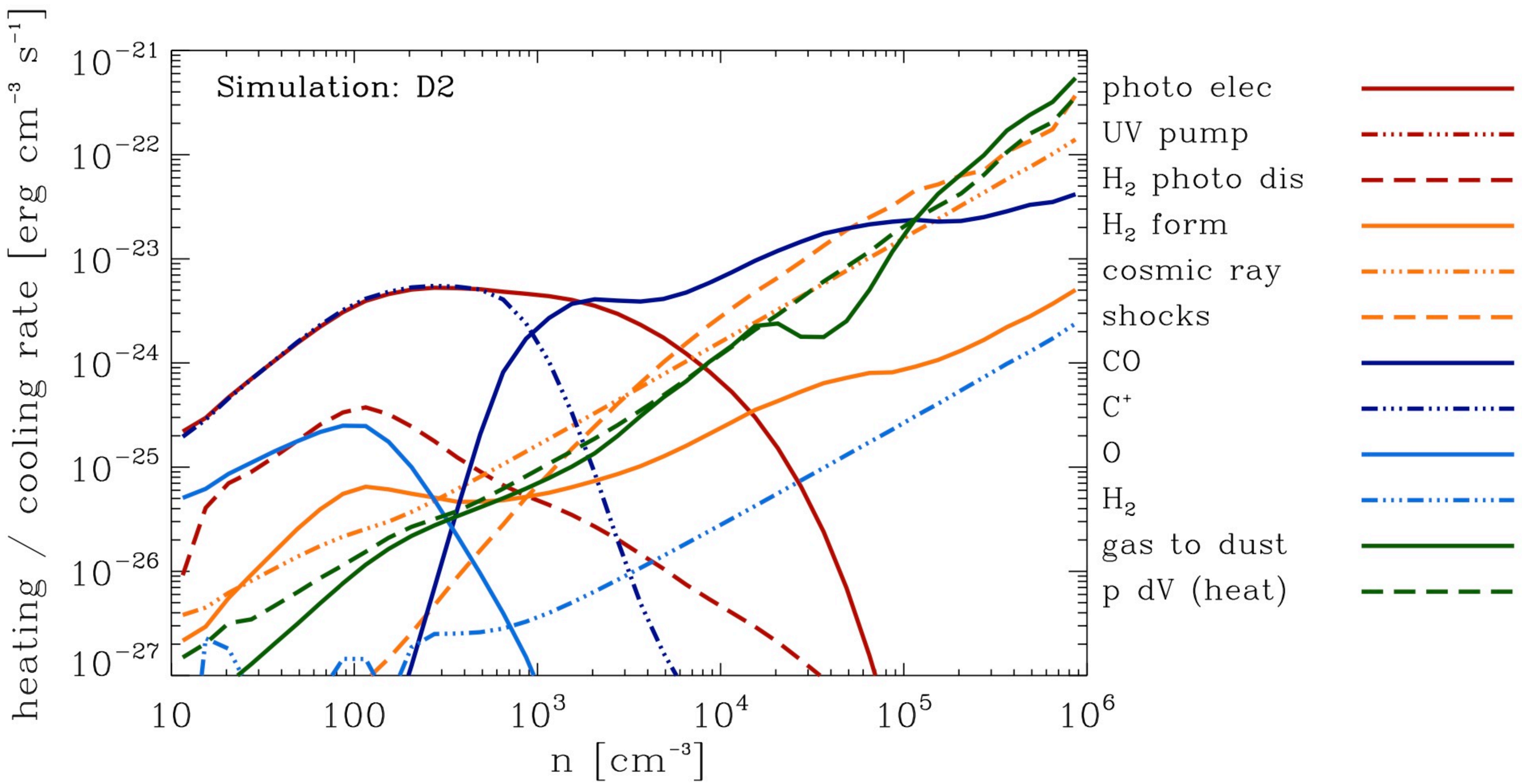
- Improvements:
  - Follow ionisation state of H explicitly, only assume equilibrium for the metals
  - Assume different equilibrium curves for photoionised, collisionally ionised gas and interpolate between them

Solid line: collisional ionisation equilibrium
Dashed line: photoionisation equilibrium



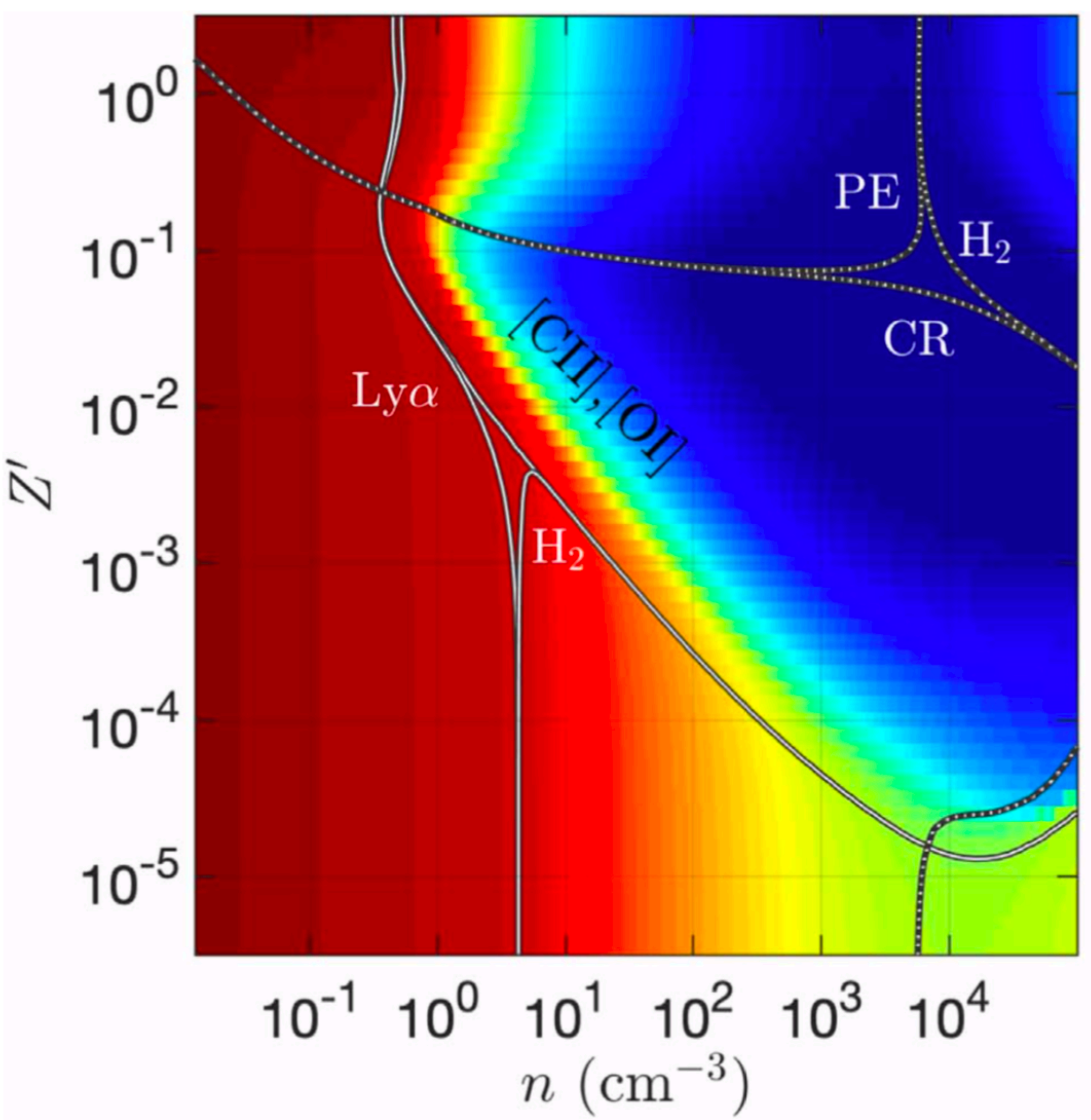
Kim et al. (2023)

- Below 10000 K, cooling dominated by atomic fine structure lines, molecules, dust



Glover & Clark (2012)

- At solar metallicity, CO dominates only within ~1 order of magnitude; H_2 almost never significant
- At low metallicity, CO almost never important, but H_2 starts to play more significant role



Bialy & Sternberg (2019)

- Properly modelling cooling in this regime requires simultaneous treatment of chemistry
- For accurate cooling, minimal model needs to follow ionisation balance of H, C, plus H/H_2, $C^+/C/CO$ transitions
- Many models do not do this
  - Impact on dynamics unclear
  - Definitely important if making direct comparison of atomic/molecular emission with observations

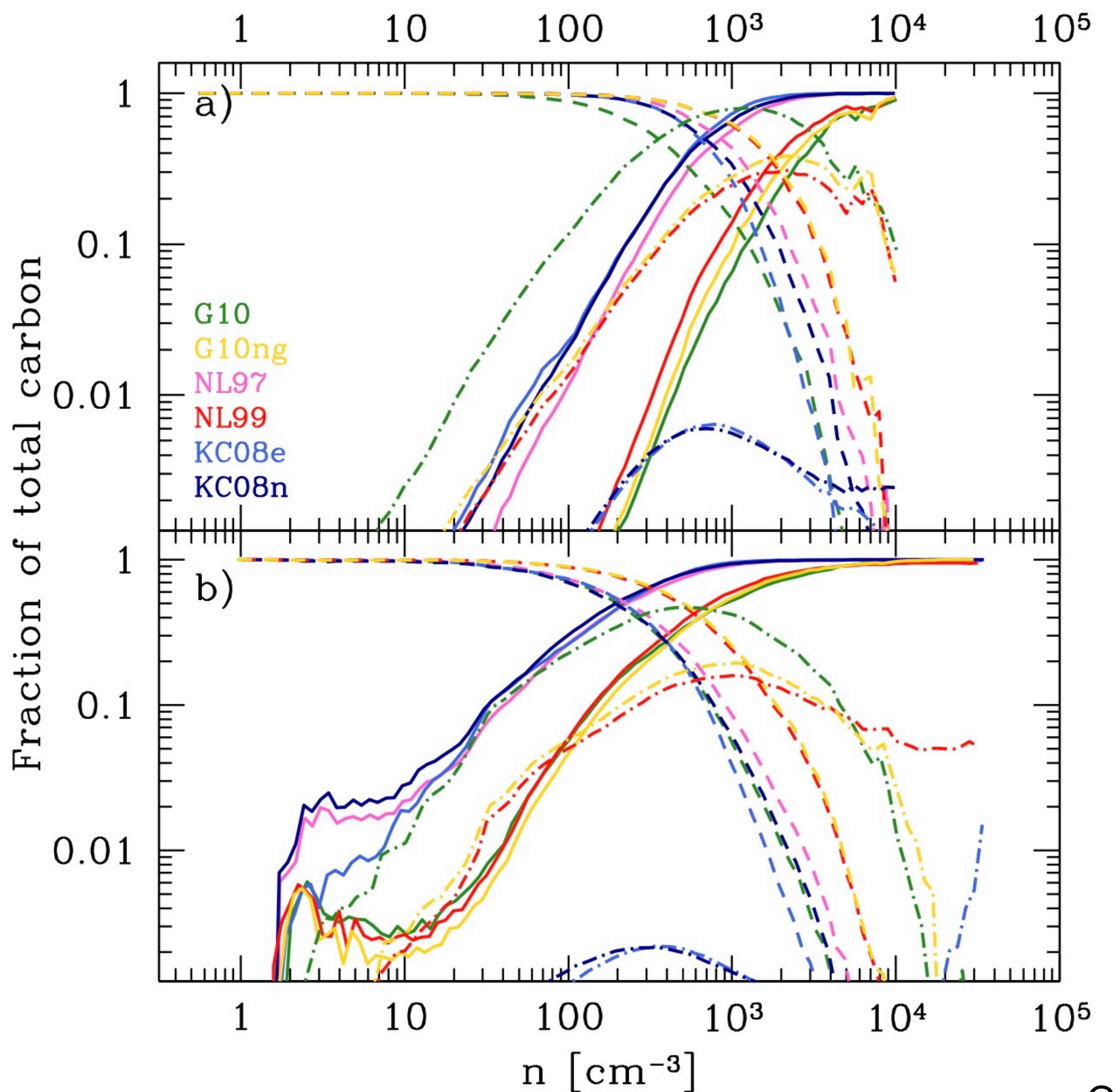
Chemistry

- ISM is chemically rich
  - Over 100 molecules detected in ISM
  - State-of-the-art astrochemical models contain 1000s of reactions
- Modelling this in large 3D simulations is impractical
  - Rate equations are stiff, require implicit solution
  - For N species, cost scales as N^3

- Solution: reduced chemical network
  - Follow chemistry vital for cooling
  - Small subset of full chemistry — major speedup!
  - Other species can be modelled later, in post-processing, **if right information is saved**
  - What do we need for post-processing?
    - Density, temperature, angle-averaged A_V for set of Lagrangian fluid elements

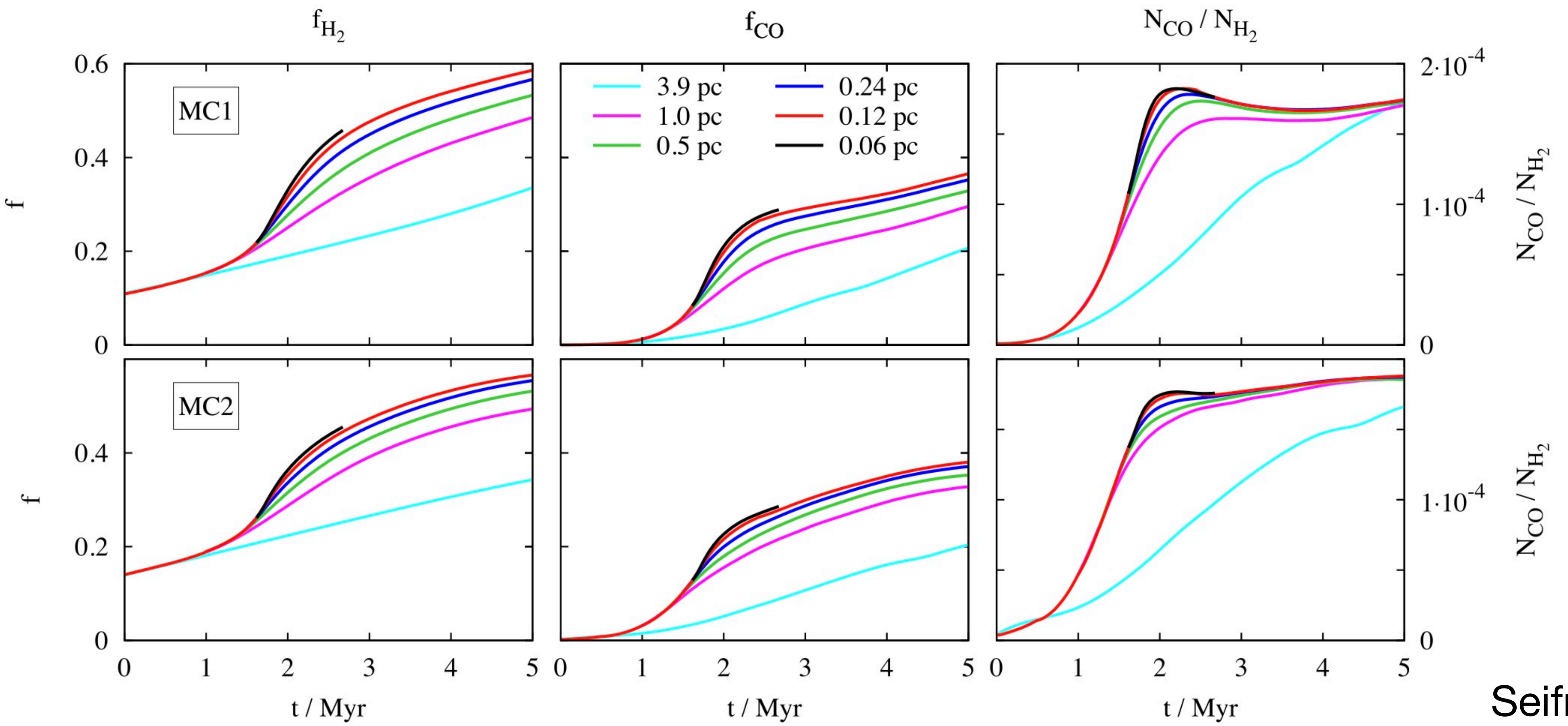
- Various simple networks available:
  - Only hydrogen (H^+, H, H_2)
  - Nelson & Langer (1997): hydrogen, extremely simplified CO [3 reactions, no atomic C]
  - Nelson & Langer (1999): hydrogen, more accurate CO chemistry
  - Gong et al. (2017): improved version of NL99, more accurate when cosmic ray ionisation rate large

- NL97 network is extremely cheap, but can make large errors in CO abundance



Glover & Clark (2012)

- Accurate modelling of C+/C/CO transition needs high spatial resolution



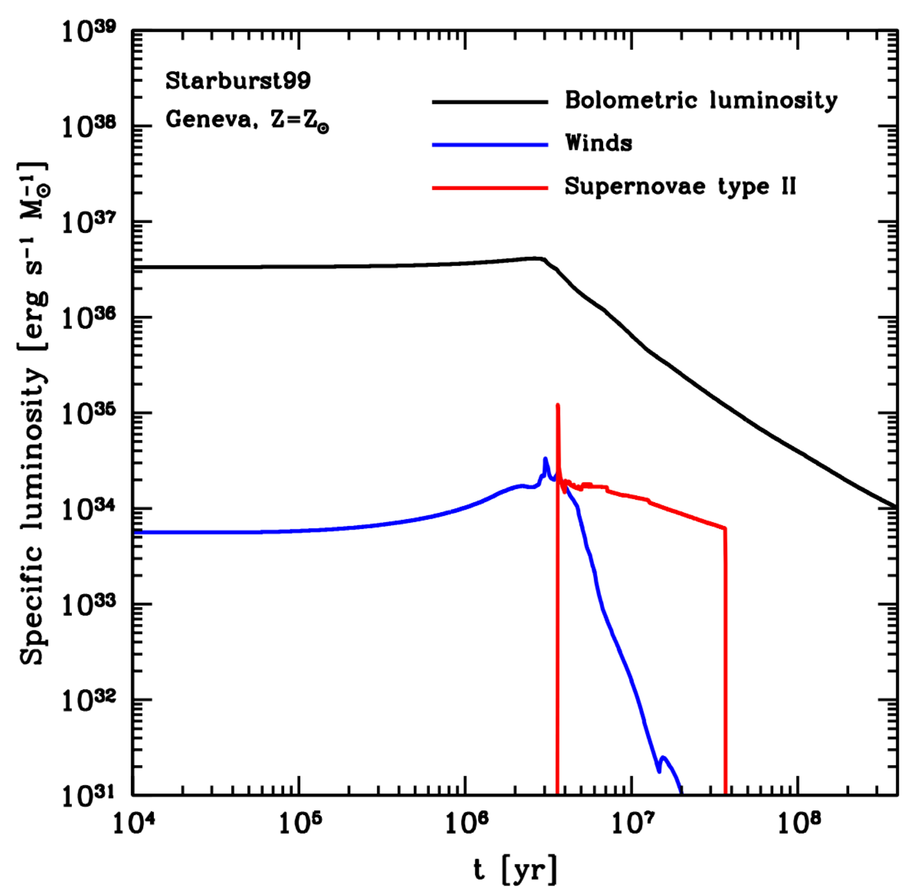
Seifried et al. (2015)

- NL97 good choice for low resolution simulations, where we don't expect high accuracy

- For high resolution sims, use Gong et al. (2017)

Stellar feedback

- On small scales (<~ pc), protostellar jets important, but impact highly localised — not important on large scales
- On large scales (>> pc), feedback from massive stars dominates total feedback energy, momentum budget



Agertz et al. (2013)

Supernovae (SNe)

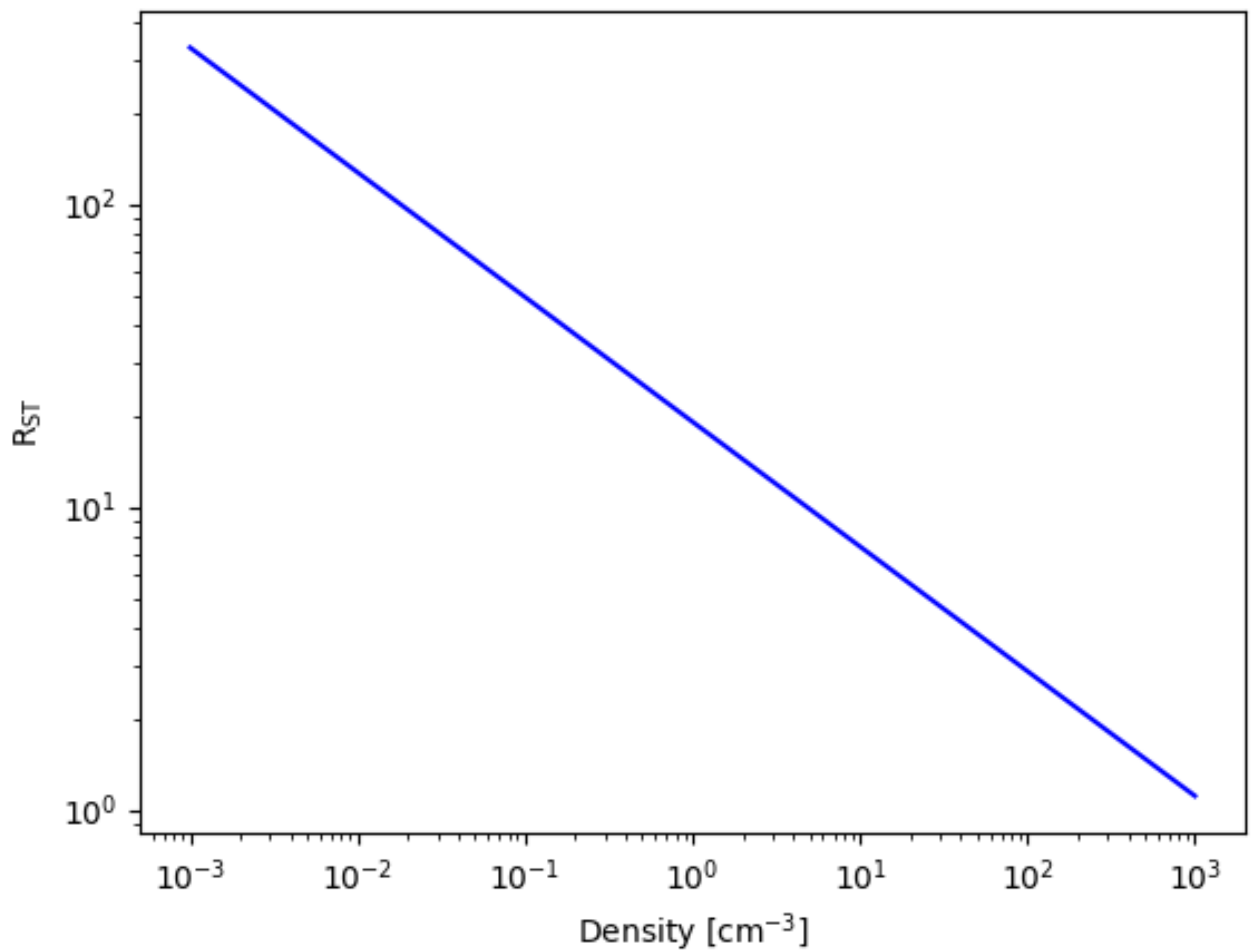
- Simplest way to model SNe in large-scale simulations is by depositing SN energy as thermal energy
- Problem: if local gas density too high, energy can be radiated away before low density bubble forms
- **Overcooling** leads to inefficient feedback, excessive star formation in galaxies
- Fundamentally, a resolution problem!

- For SNe exploding in uniform density gas, we can predict size of remnant once cooling starts to be important (R_{ST} — end of Sedov-Taylor phase)

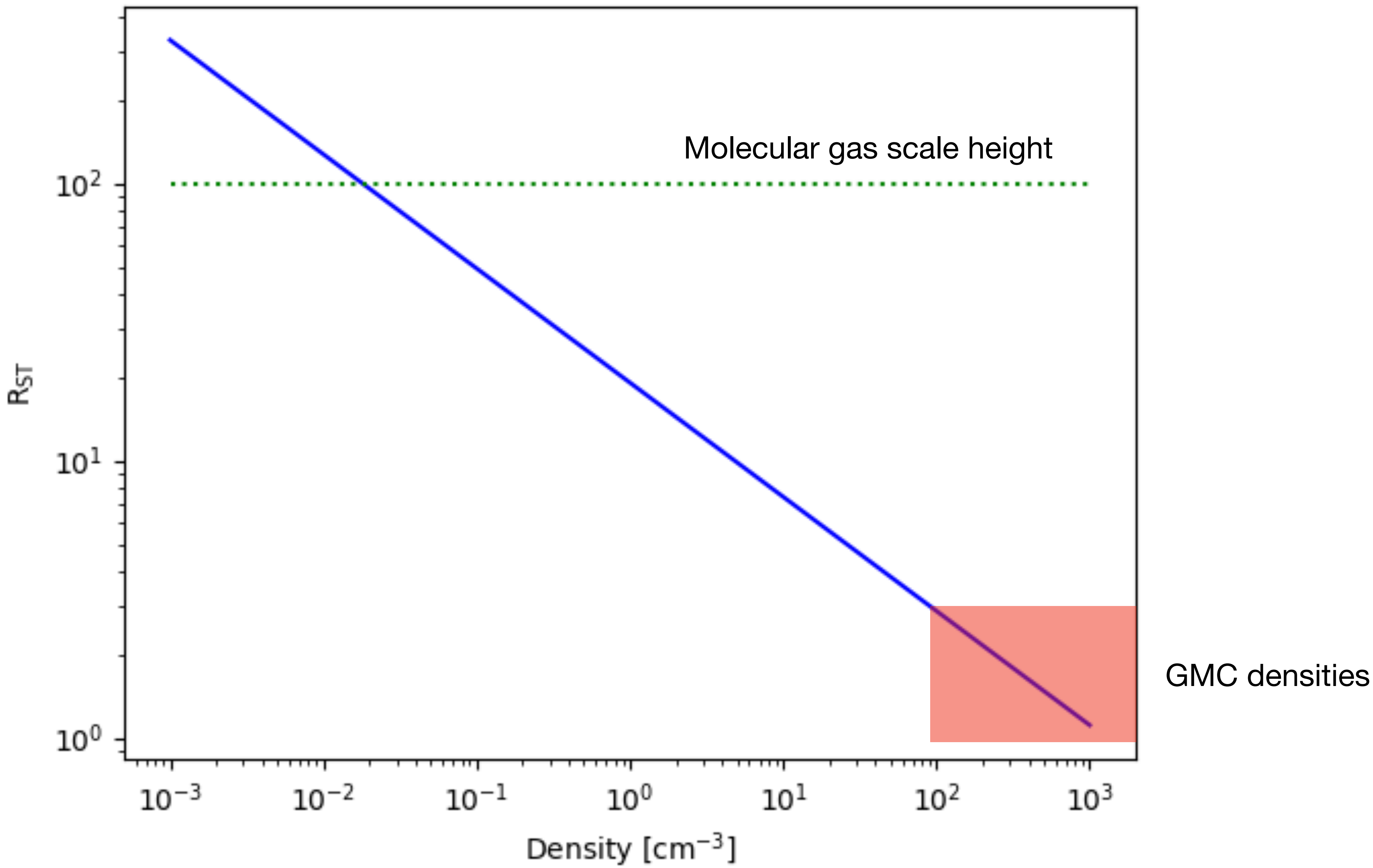
- If SN well-resolved at this point, then no overcooling

- Compare resolution, R_{ST} at moment SN explodes
  - If R_{ST} well-resolved, inject energy
  - If R_{ST} not well-resolved, inject appropriate terminal momentum

- When running with chemistry, also need to think about updating chemical state to be consistent with new local temperature!



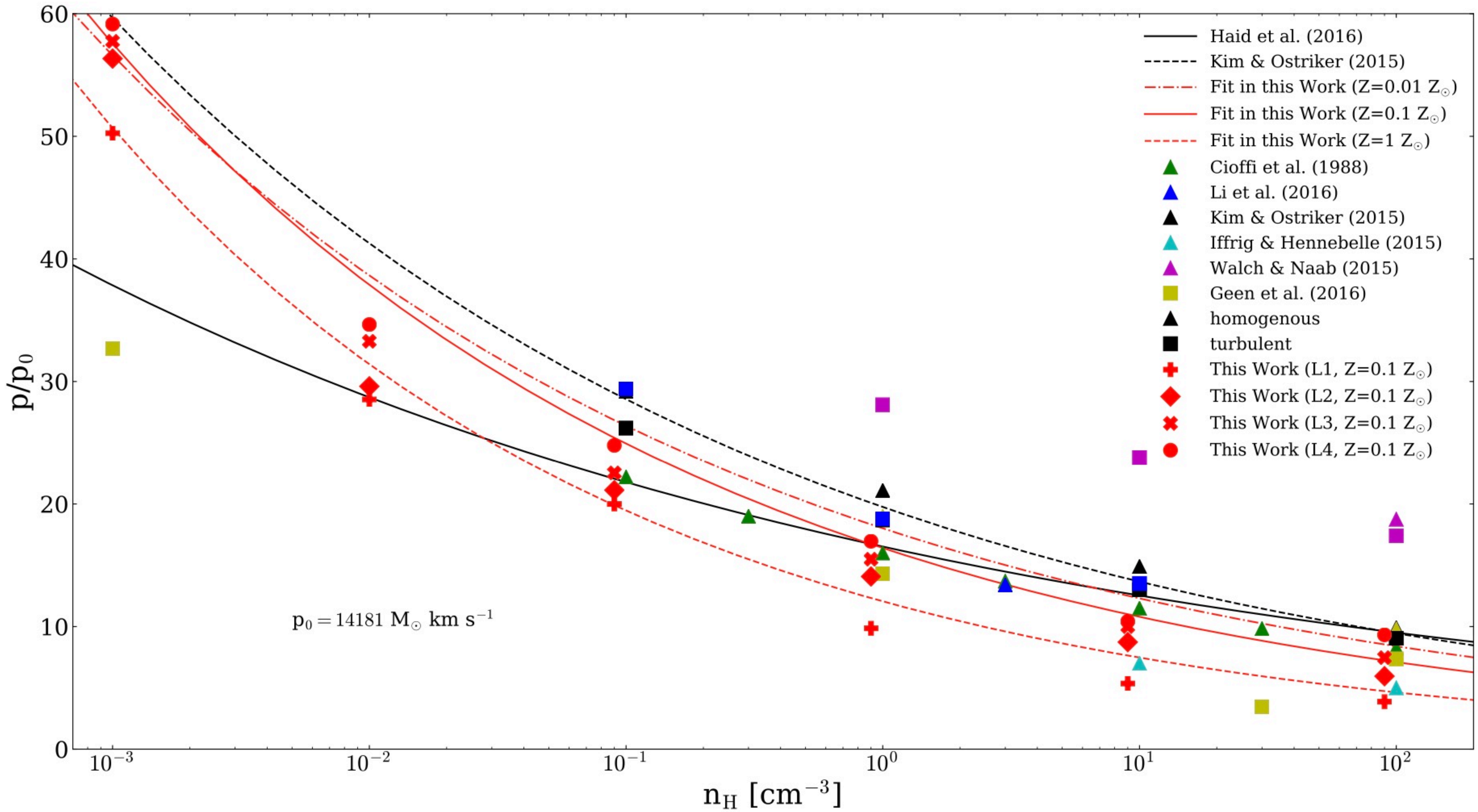
R_{ST}
10^2
10^1
10^0
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3
Density [cm^{-3}]


Molecular gas scale height
GMC densities
R_ST
Density [cm⁻³]
10^2
10^1
10^0
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3

- Sub-pc resolution needed to resolve energy injection at typical GMC densities — difficult to afford!

- R_{ST} becomes very large at very low densities
  - Injecting energy within sphere of radius R_{ST} can become problematic when density very low
  - Issue worse in Lagrangian codes — low spatial resolutions at low densities (because little mass)

- How much momentum should we inject?



Hu et al. (2019)

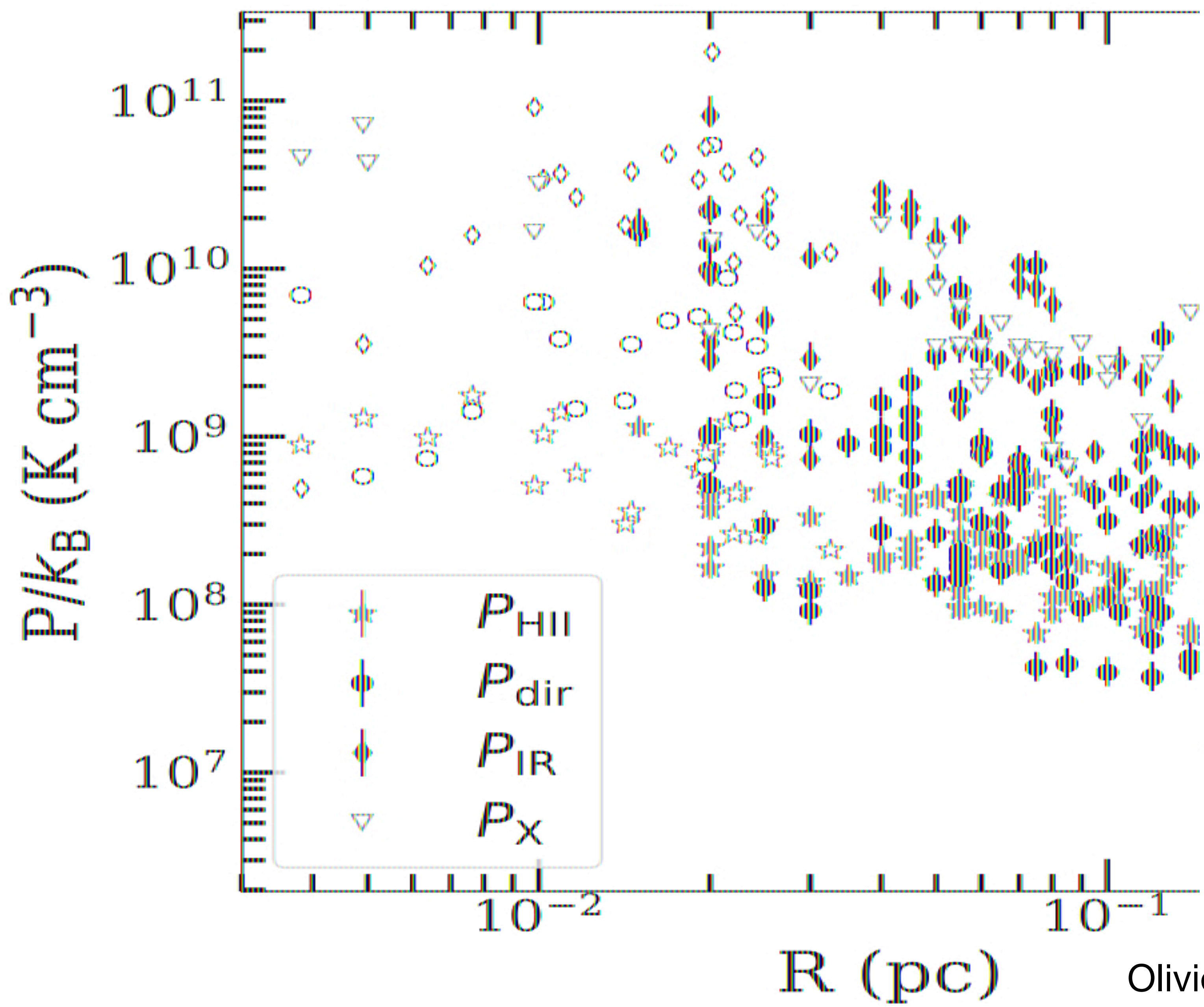
- Where do we inject the energy/momentum?
  - Completely randomly?
  - Randomly within small region around star particle?
  - At location of star particle?
- Choice here makes huge difference to effectiveness of SNe, since controls density in which most SNe explode
- Clustered SNe much more effective at disrupting disk!

- With SN feedback alone, simulated ISM differs from real one in several important respects:
  - Too large a volume of hot gas
  - Too much mass in cold gas
  - Too much star formation
  - Star cluster masses too large
- All of these issues have same root cause: too much gas collapse and star formation can occur before first SNe

“Early” feedback

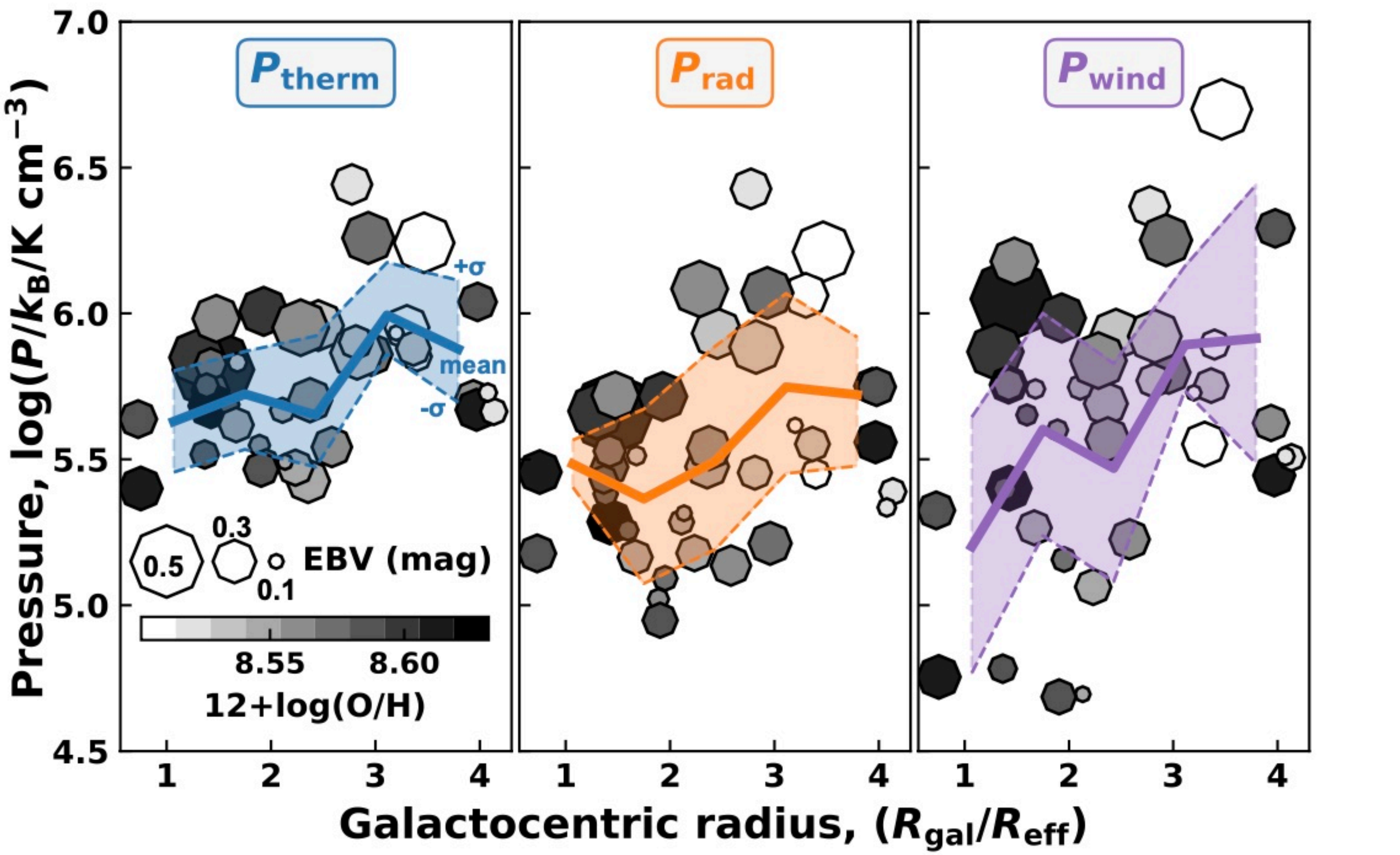
- SN feedback only acts once massive stars reach the end of their lives — time delay of ~ 4 Myr at solar Z
- Plenty of time for clouds to collapse, stars to form before SN feedback becomes active
- Solution: Include some form of “early” feedback that acts as soon as massive stars reach main sequence
- Leading candidates: winds, ionising radiation

- On very small scales, energy-driven winds, radiation pressure dominate

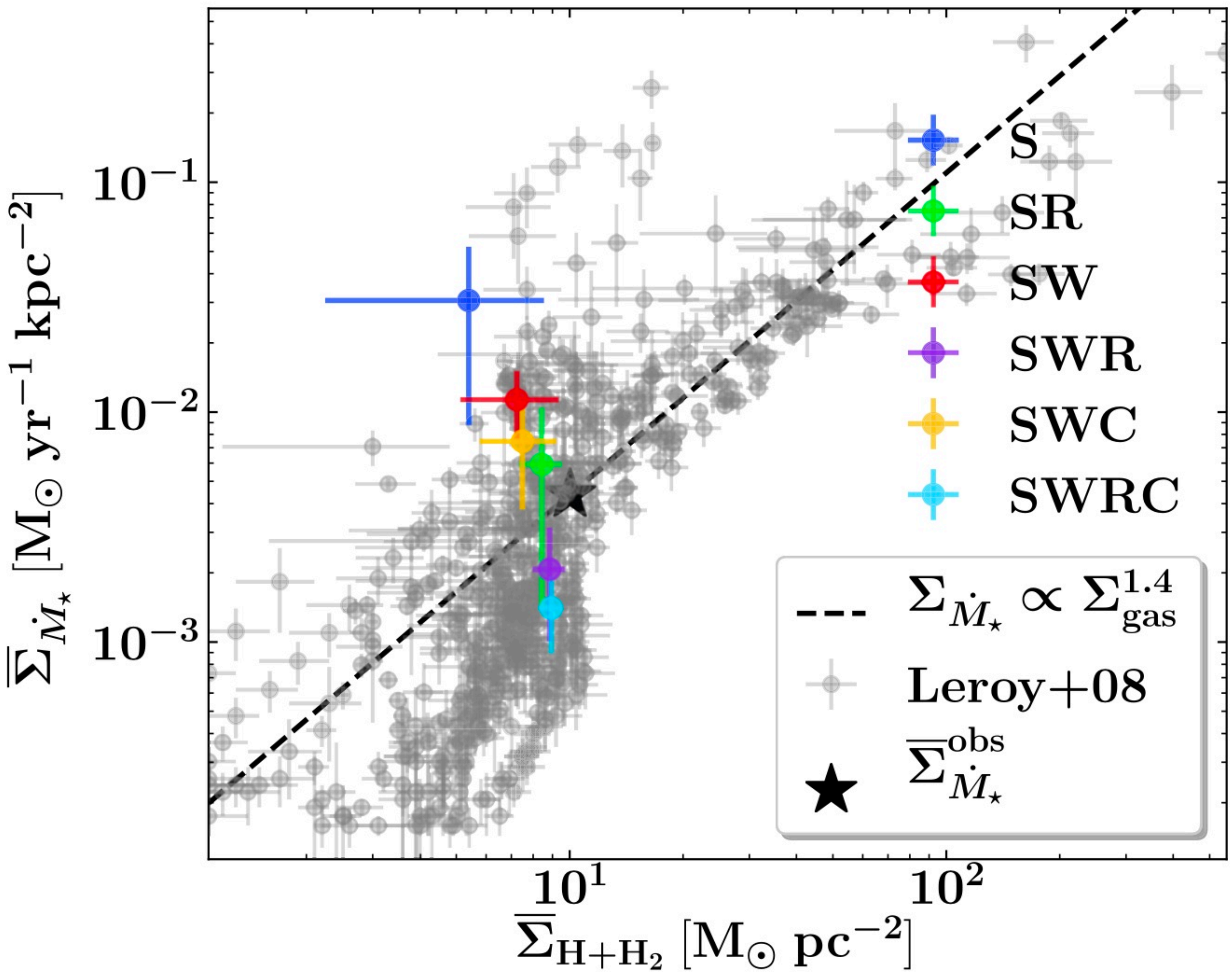


Olivier et al. (2021)

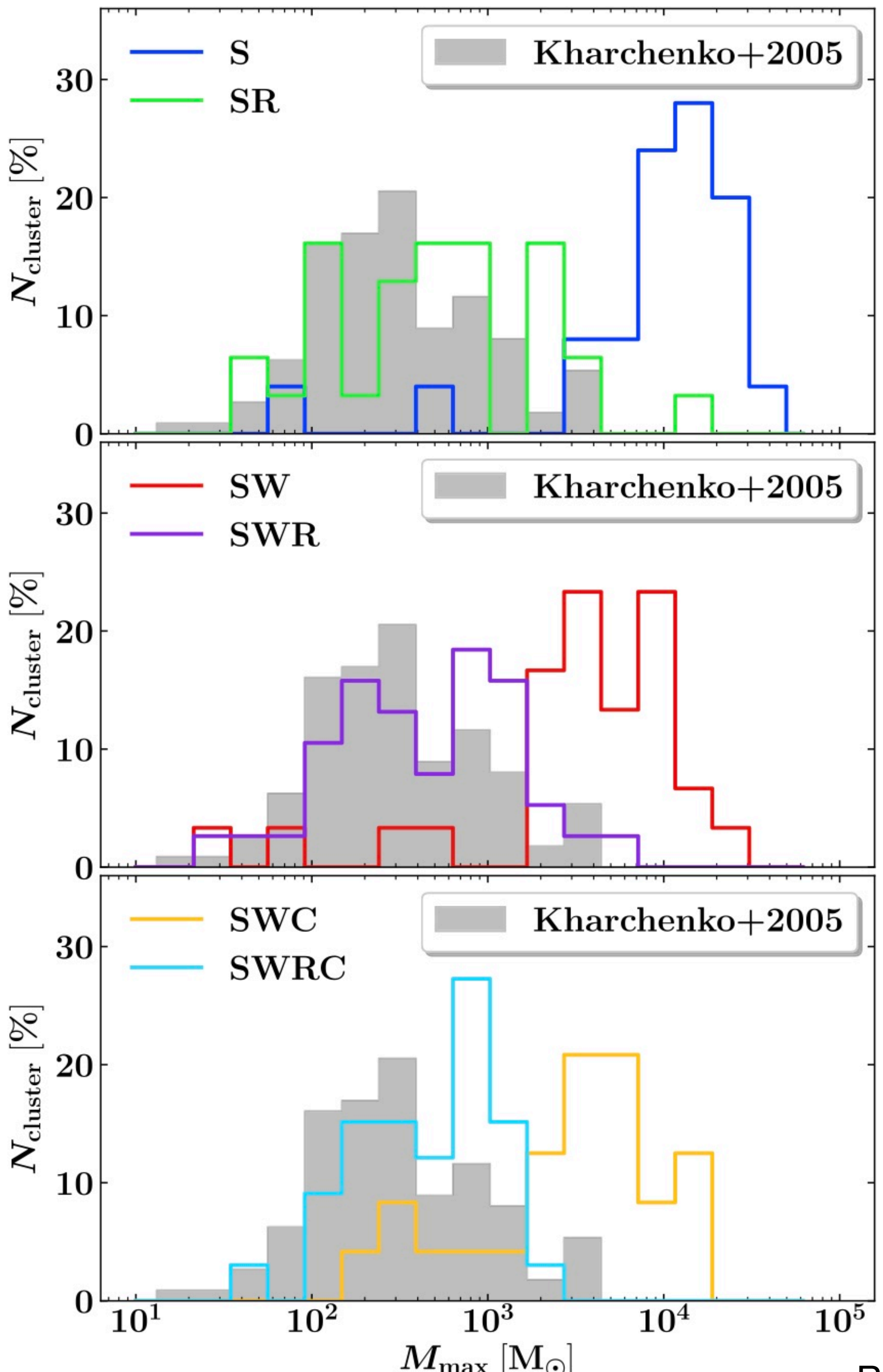
- On large scales, photoionised gas pressure dominates (but winds, radiation pressure not negligible…)



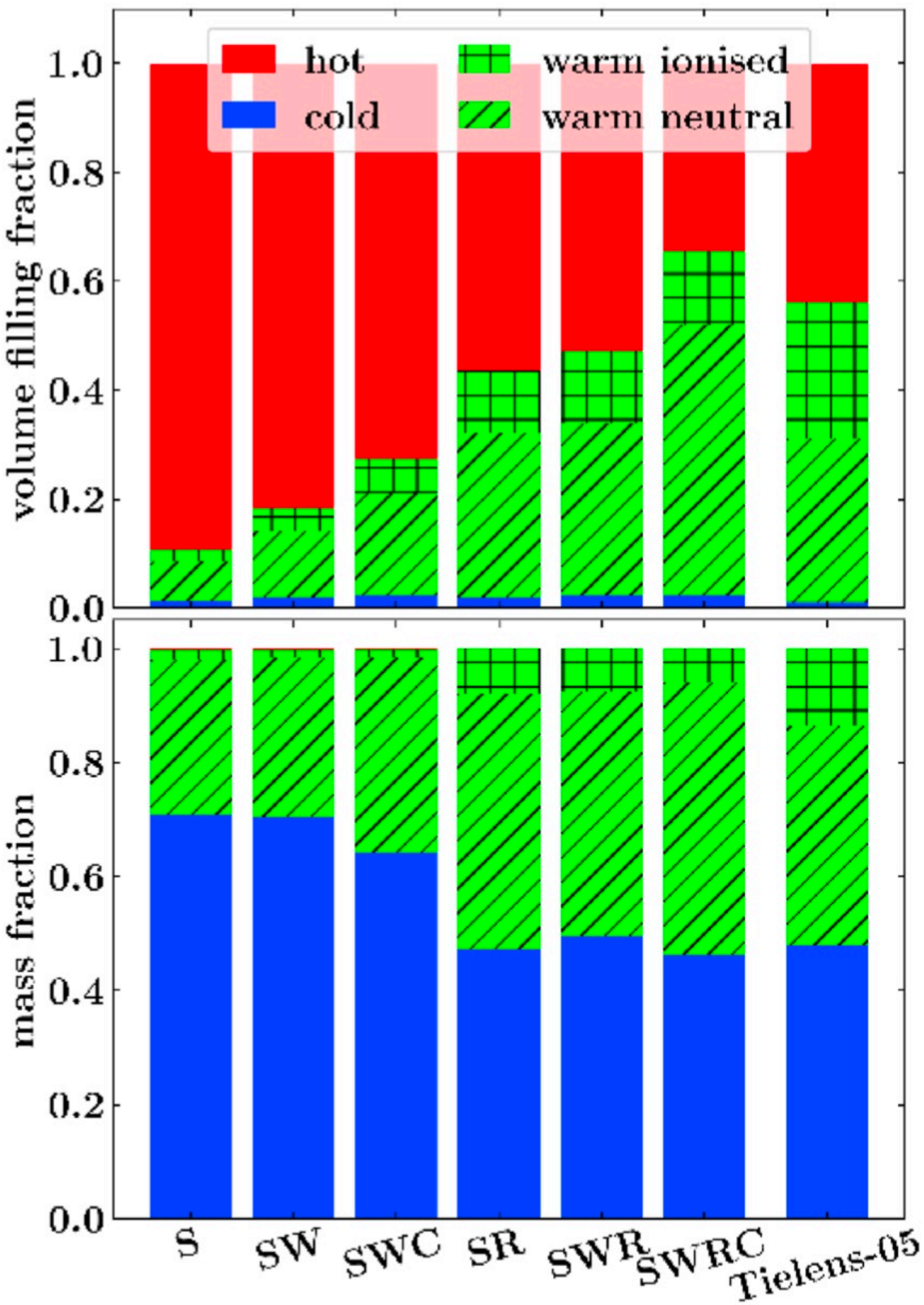
Barnes et al. (2022)



Rathjen et al. (2021)



Rathjen et al. (2021)



Rathjen et al. (2021)

- Best match to observations when we include all relevant forms of early feedback

- If this isn’t affordable/practical, then SNe + radiation much better than SNe alone

- Inclusion of radiation also crucial for predictions of many observables (e.g. Hα emission)

Cosmic rays

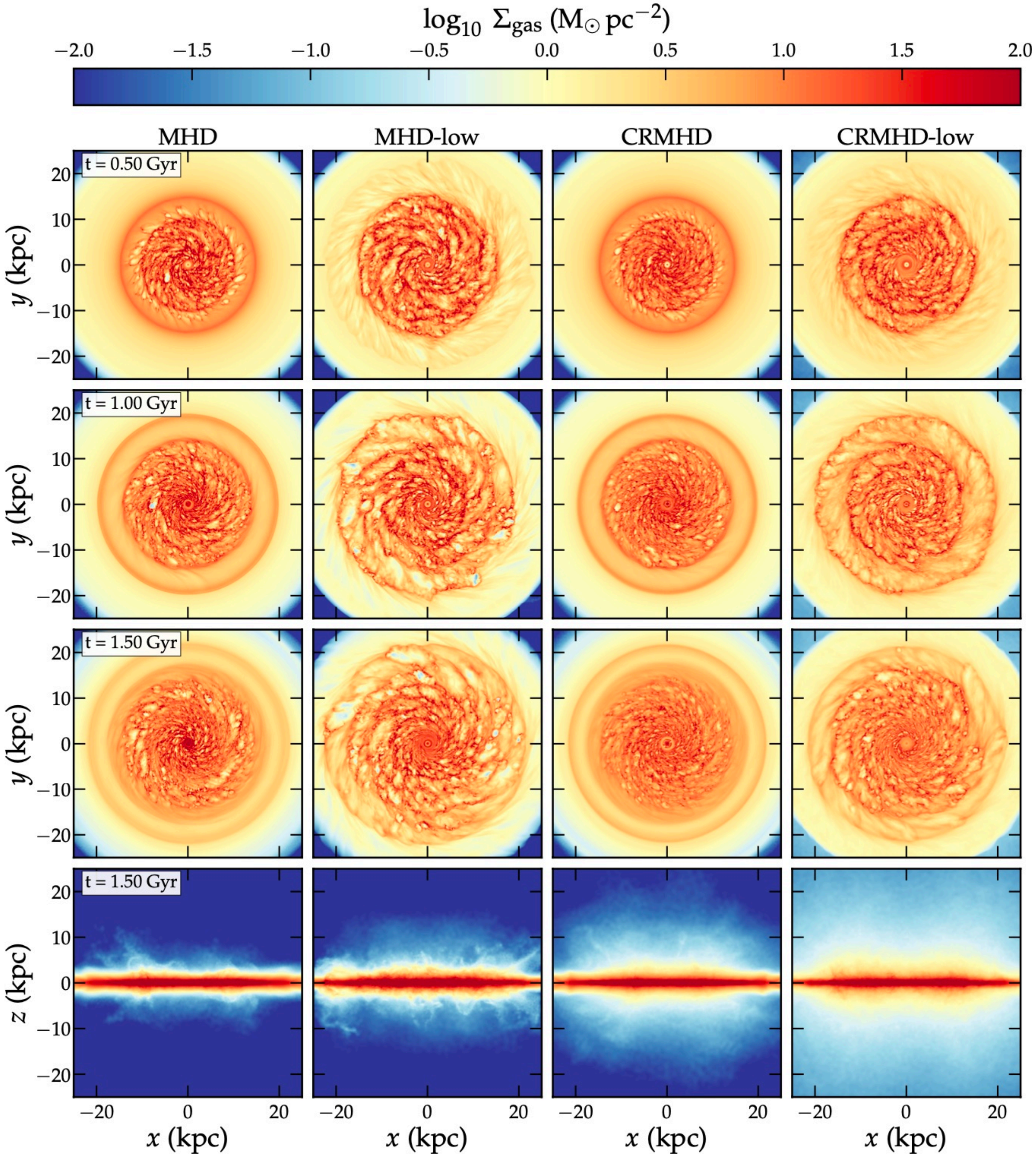
- Energy density in cosmic rays comparable to that in B field, turbulence, gas thermal energy
- Cosmic rays potentially play important role in regulating large-scale dynamics of ISM
- Despite this, their behaviour in large ISM simulations remains relatively under-explored

- Simple approach: adopt gray approximation, model cosmic rays as single fluid with γ = 4/3

- Account for production in SNe, losses due to interaction with gas, B fields

- Also need to account for diffusion along field lines

$$
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \\
&\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{v} \boldsymbol{v}^{\mathrm{T}} + P \mathbf{I} - \frac{\boldsymbol{B} \boldsymbol{B}^{\mathrm{T}}}{4\pi} \right) = 0 \\
&\frac{\partial e}{\partial t} + \nabla \cdot \left[(e + P) \boldsymbol{v} - \frac{\boldsymbol{B} (\boldsymbol{v} \cdot \boldsymbol{B})}{4\pi} \right] = \\
&P_{\mathrm{cr}} \nabla \cdot \boldsymbol{v} - \boldsymbol{v}_{\mathrm{st}} \cdot \nabla P_{\mathrm{cr}} + \Lambda_{\mathrm{th}} + \Gamma_{\mathrm{th}} \\
&\frac{\partial e_{\mathrm{cr}}}{\partial t} + \nabla \cdot (e_{\mathrm{cr}} \boldsymbol{v} + (e_{\mathrm{cr}} + P_{\mathrm{cr}}) \boldsymbol{v}_{\mathrm{st}} - \kappa \boldsymbol{b} (\boldsymbol{b} \cdot \nabla e_{\mathrm{cr}})) = \\
&- P_{\mathrm{cr}} \nabla \cdot \boldsymbol{v} + \boldsymbol{v}_{\mathrm{st}} \cdot \nabla P_{\mathrm{cr}} + \Lambda_{\mathrm{cr}} + \Gamma_{\mathrm{cr}} \\
&\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \cdot (\boldsymbol{B} \boldsymbol{v}^{\mathrm{T}} - \boldsymbol{v} \boldsymbol{B}^{\mathrm{T}}) = 0,
\end{aligned}
$$

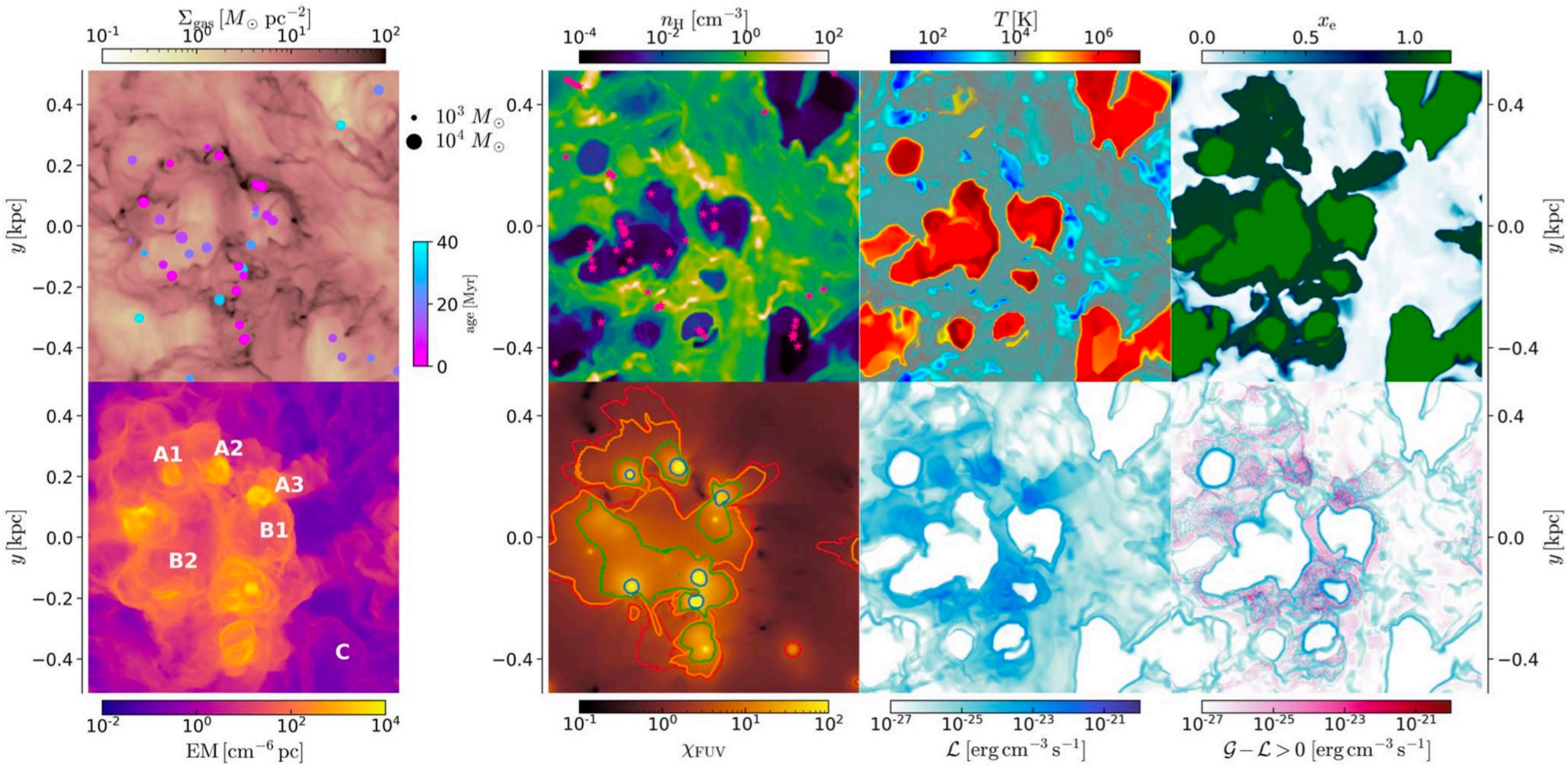


Kjellgren et al. (2025)

Examples

TIGRESS-NCR

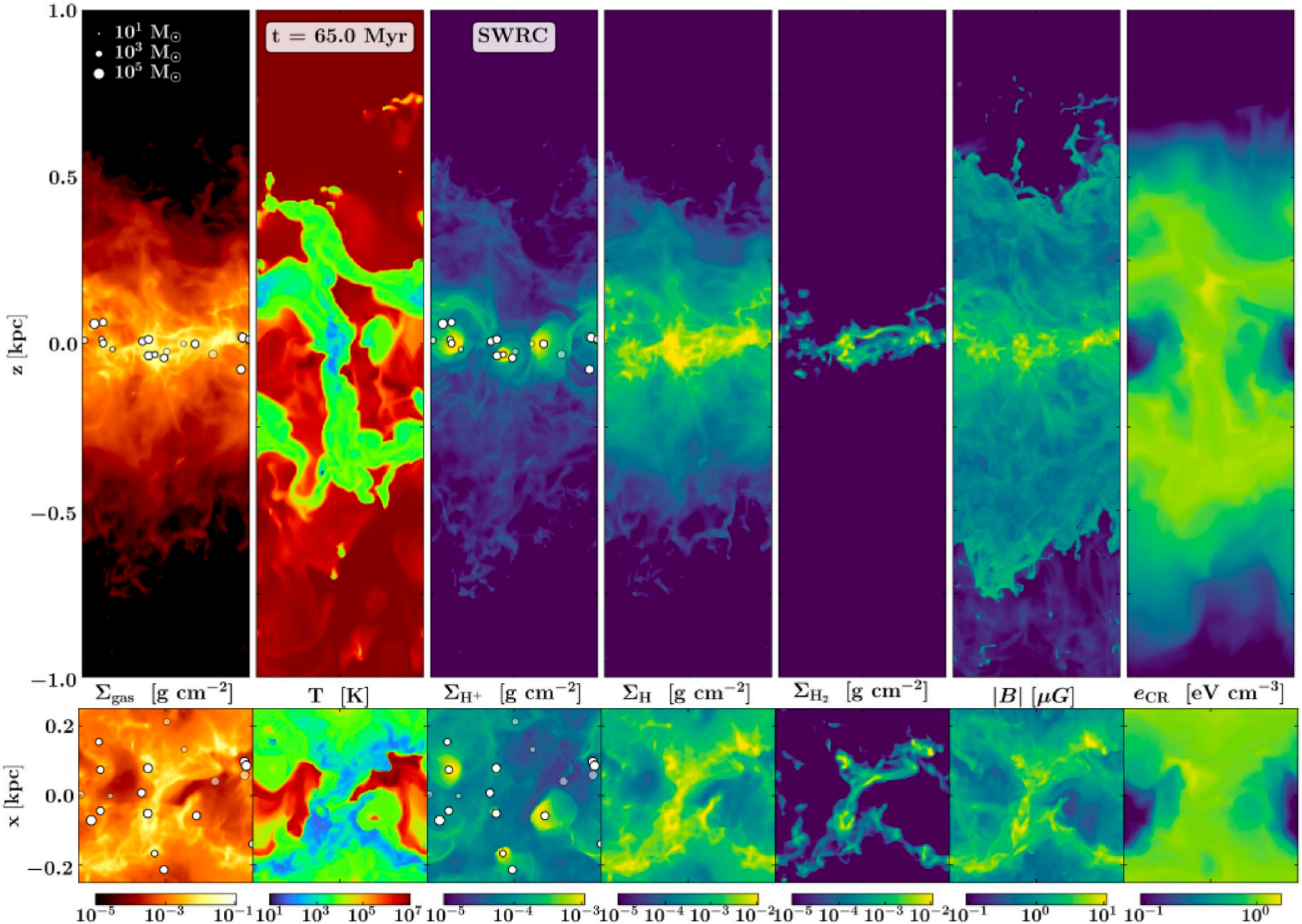
- MHD simulations of (1 kpc)^2 patch of ISM with Athena
- Eulerian, shearing sheet boundary conditions
- FFT gravity, sink particle based SF
- SNe, stellar radiation
- Detailed treatment of cooling, chemistry



Kim et al. (2023)

SILCC

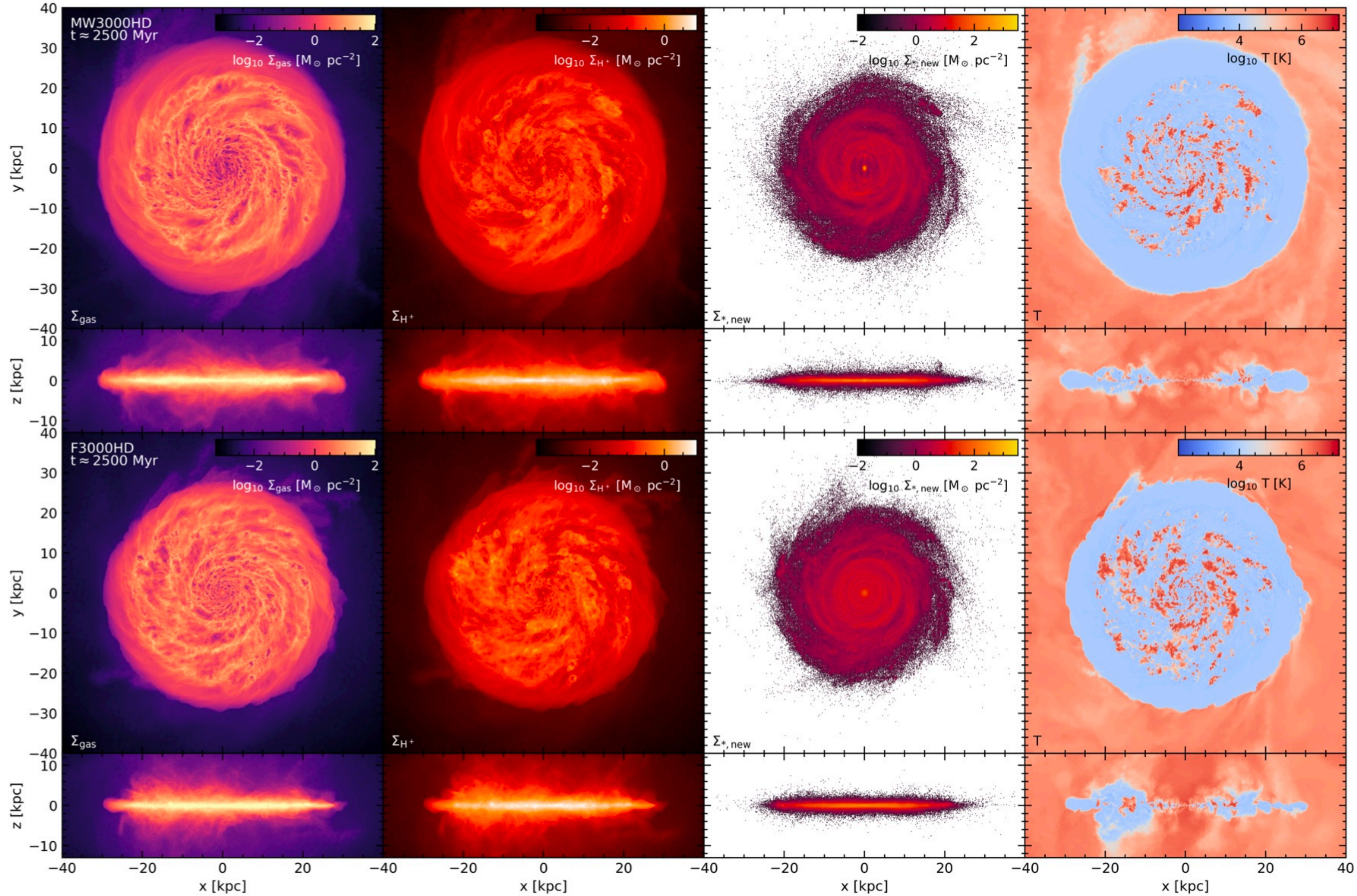
- MHD simulations of $(0.5\ \mathrm{kpc})^2$ patch of ISM with FLASH
- Eulerian, periodic BCs (no shear)
- Tree gravity, sink particle based SF
- SNe, winds, radiation, cosmic rays
- Simplified chemical network



Rathjen et al. (2021)

Rhea

- Whole-galaxy MHD simulations using Arepo
- Moving mesh, tree gravity, star particles
- Feedback only from SNe
- Detailed cooling, simplified chemistry
- No early feedback (yet!)
- Self-consistent cosmic ray transport



Göller et al. (2025)

Comparisons

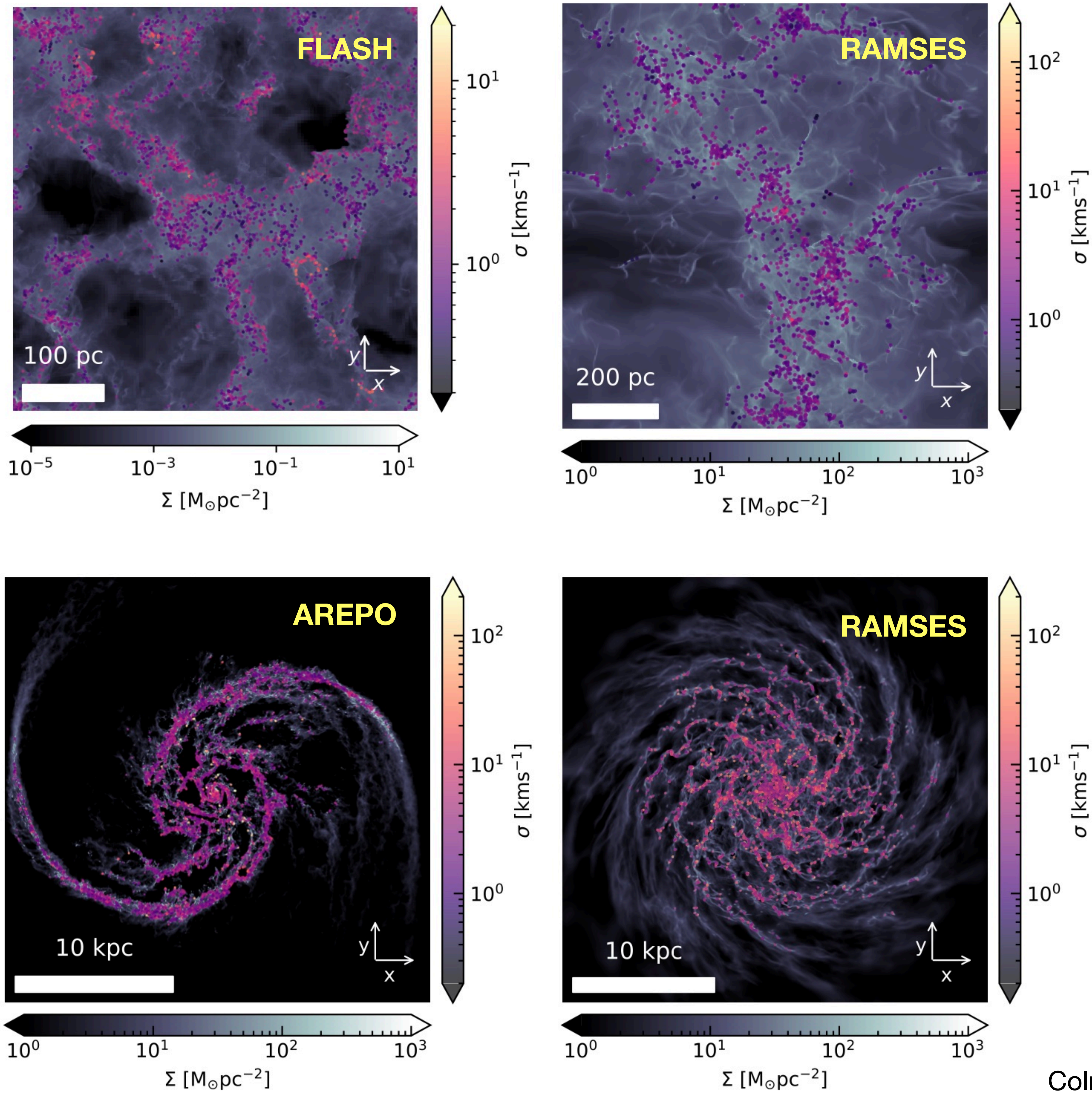
- TIGRESS-NCR:
  - Larger volume than SILCC at comparable resolution
  - Better boundary conditions, better RT
  - No cosmic rays
- SILCC:
  - Comprehensive treatment of physics, limited resolution and volume, approximate RT

- Rhea:
  - Models whole galaxy — allows direct comparison with large-scale morphology of MW
  - Worse spatial resolution, especially at low n
  - No early feedback!

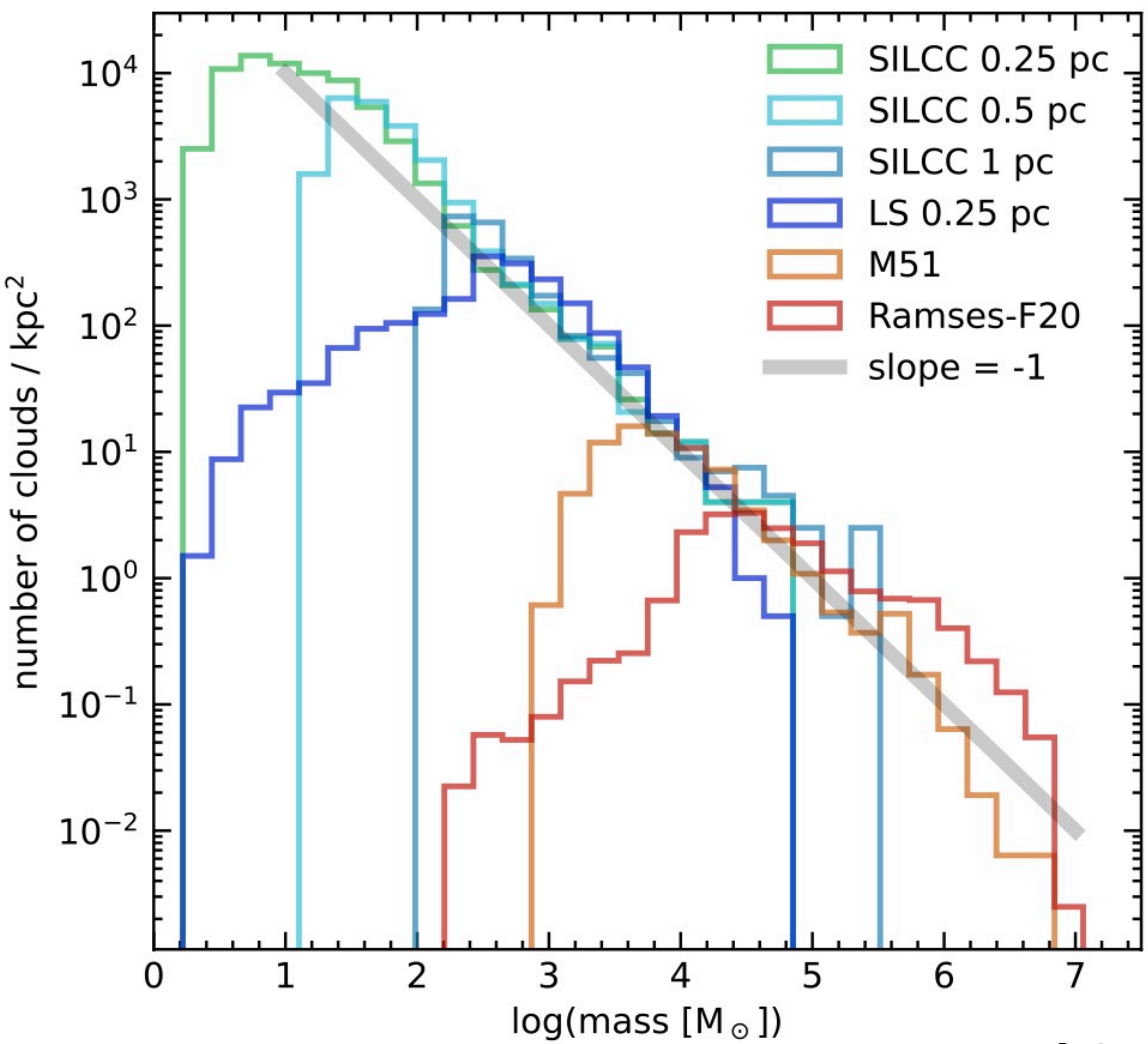
The future

- Growing number of codes available that implement most/all required physics
  - Broad range of algorithms, approximations
  - How consistent are results from different codes?

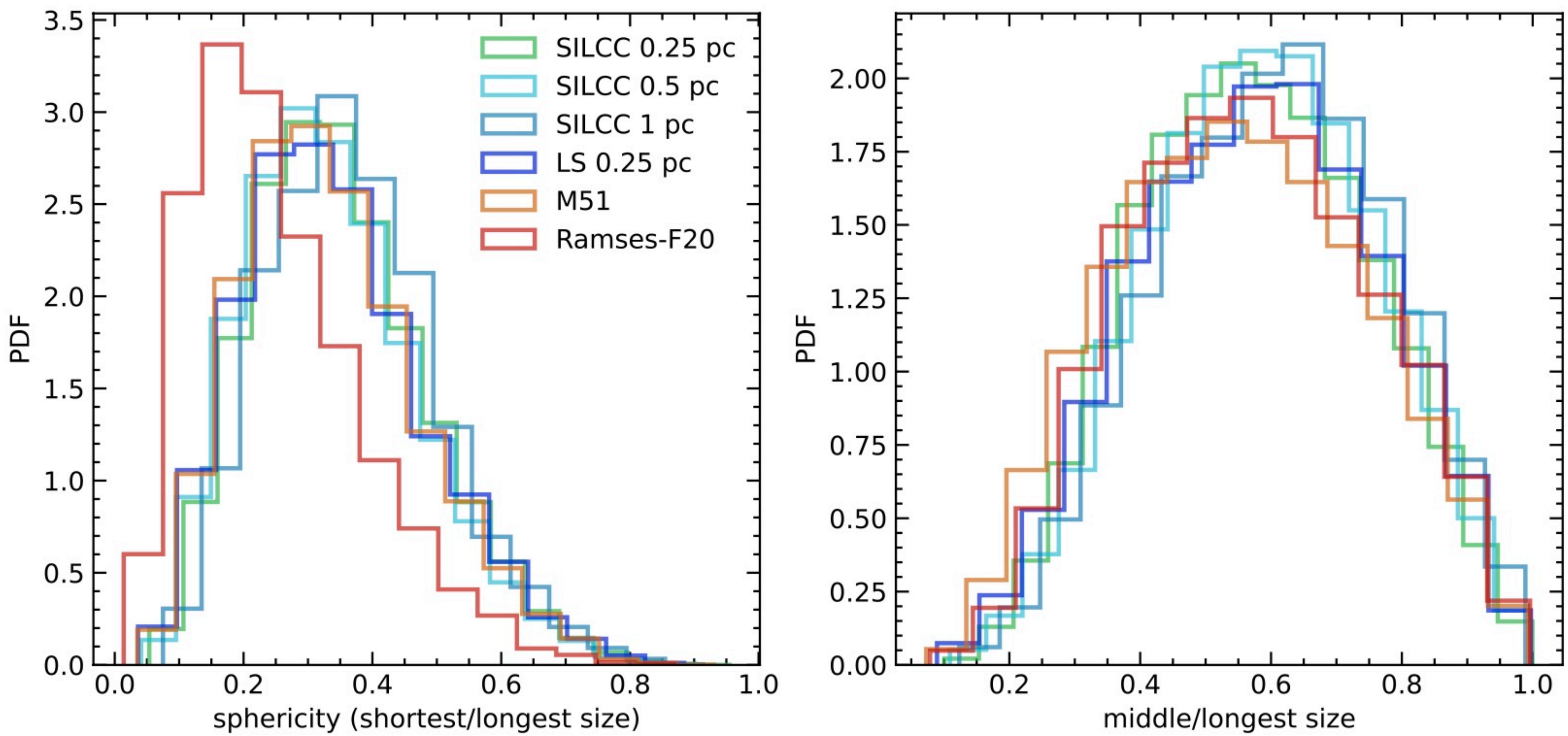
- Colman et al. (2024) compare simulations of ISM with three different codes (FLASH, AREPO, RAMSES) covering range of scales



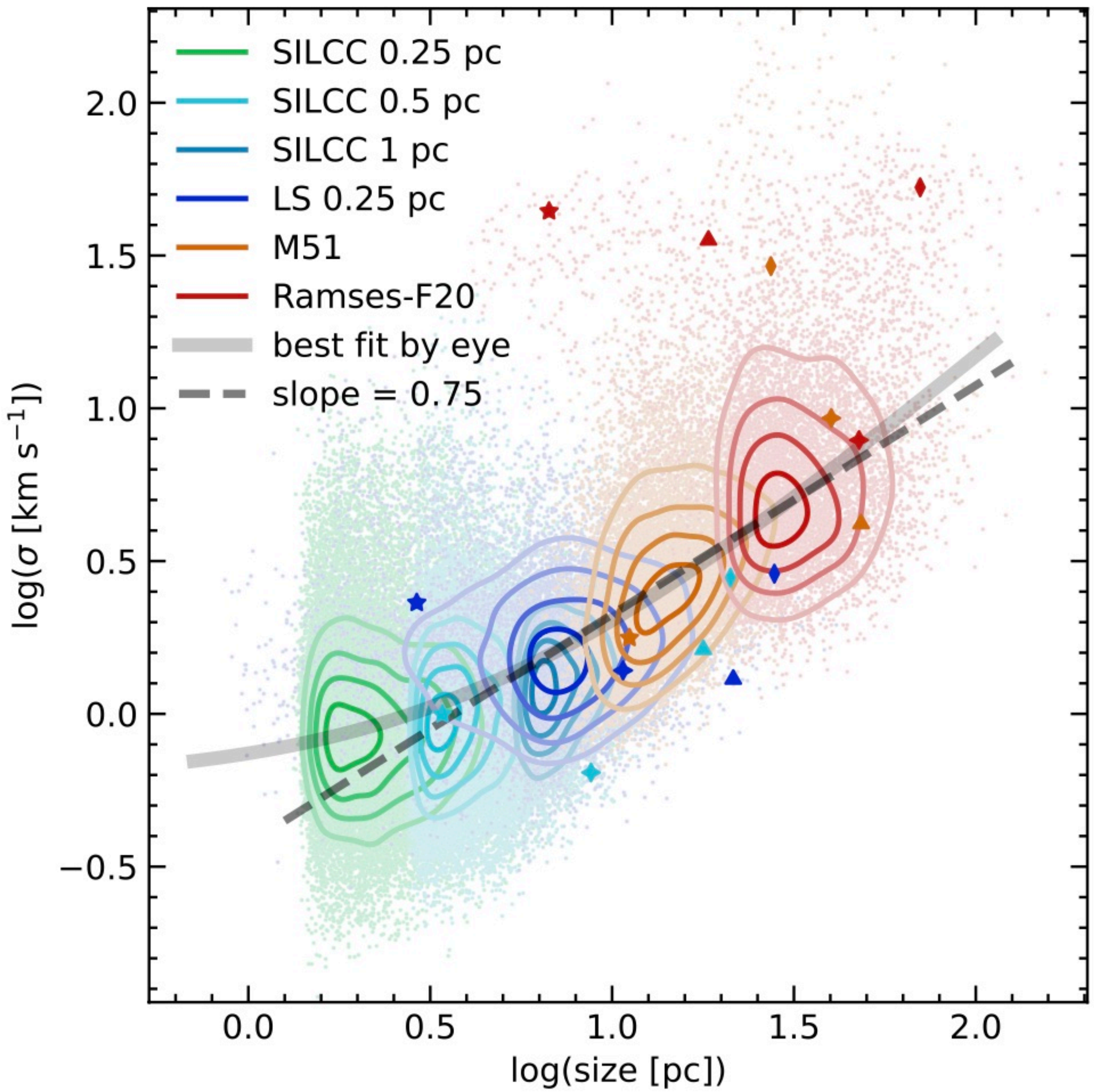
Colman et al. (2024)



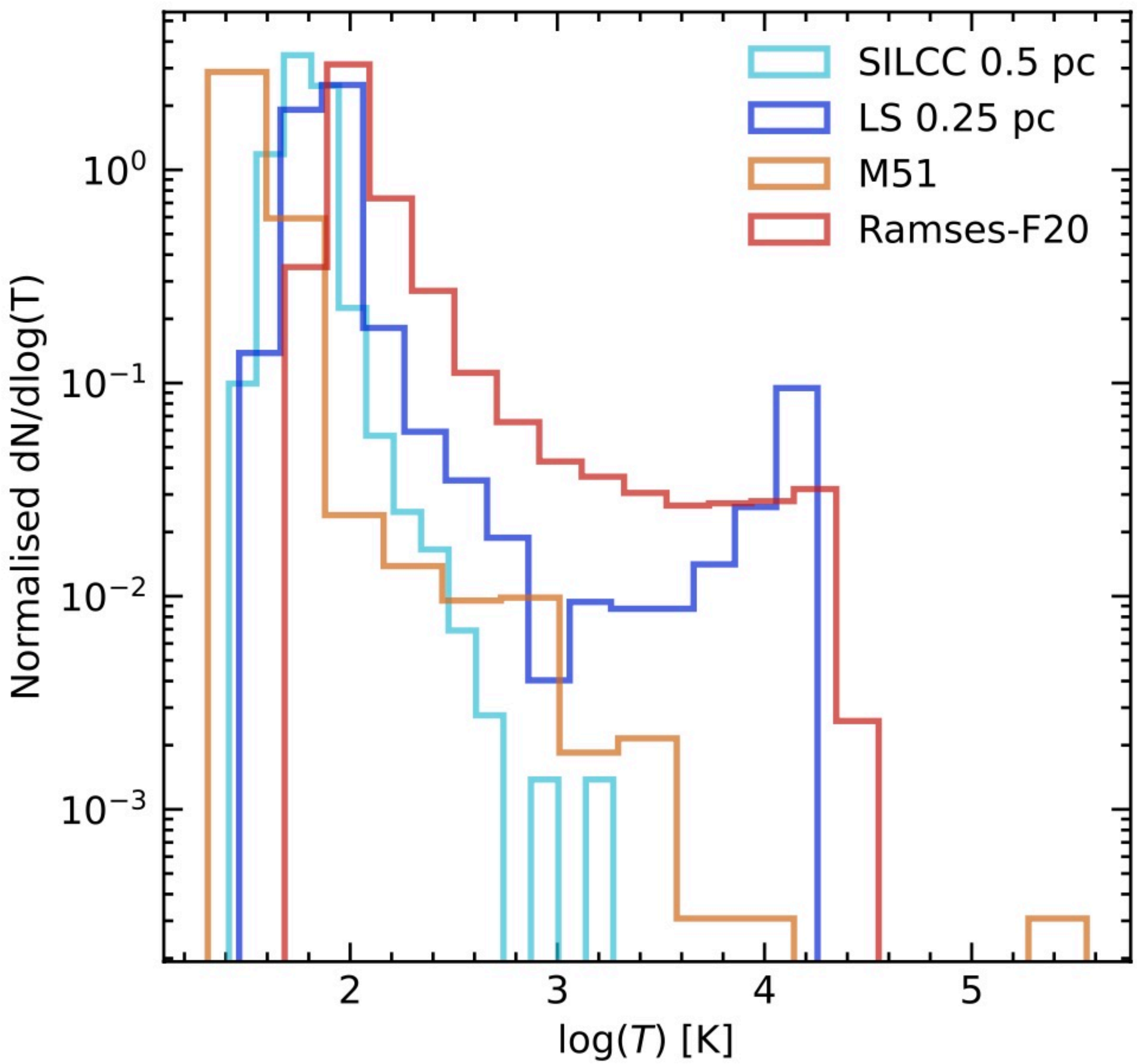
Colman et al. (2024)



Colman et al. (2024)



Colman et al. (2024)



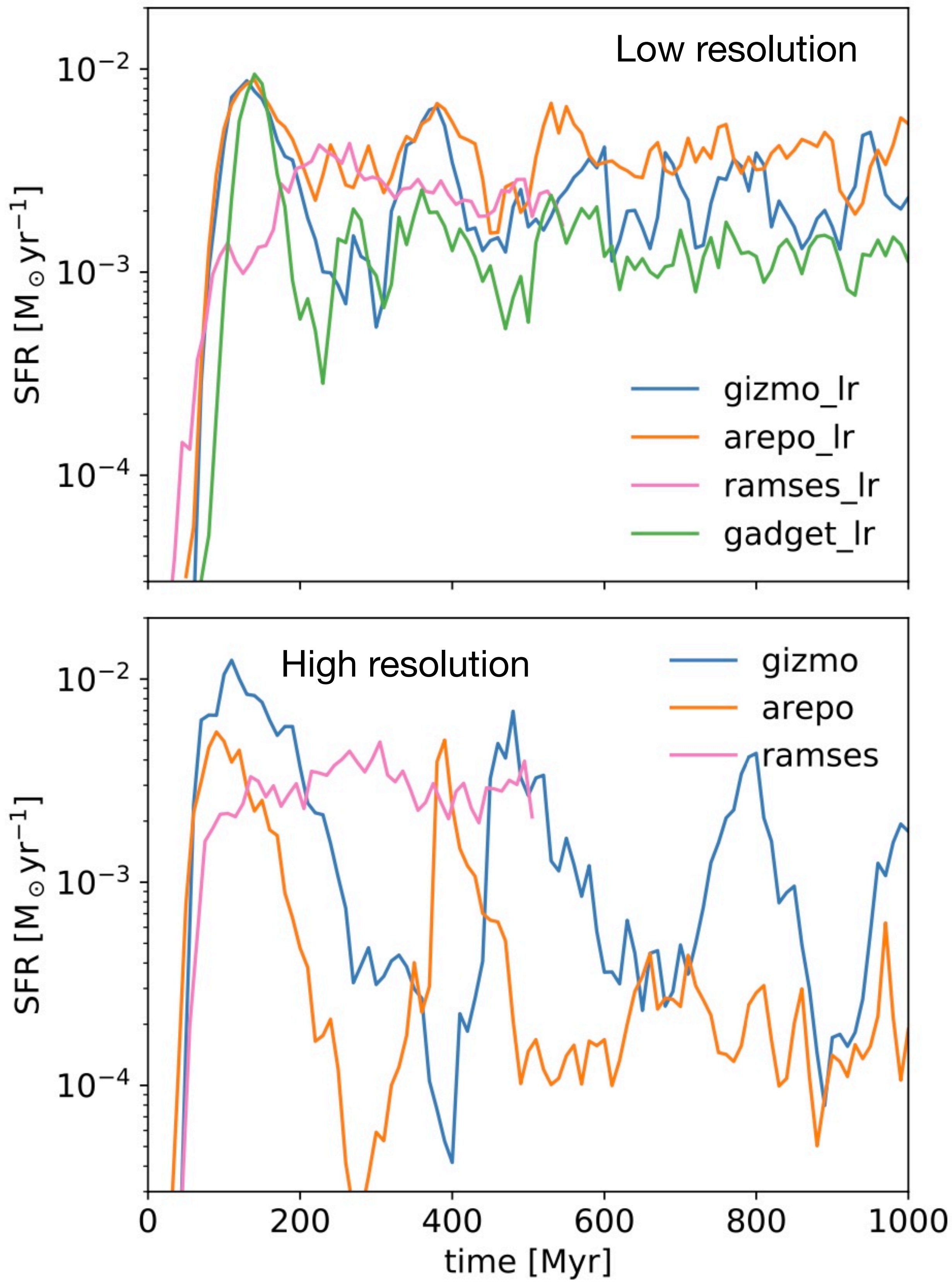
Colman et al. (2024)

- Hu et al. (2019) compare simulations of an isolated dwarf galaxy with four different codes: Gizmo, Arepo, Gadget, Ramses

- Same cooling module in all four simulations

- Star formation modelled with star particles

- Feedback purely from supernovae



Low resolution
SFR [M$_\odot$yr^{-1}]
10^{-2}
10^{-3}
10^{-4}
gizmo_lr
arepo_lr
ramses_lr
gadget_lr
High resolution
gizmo
arepo
ramses
0
200
400
600
800
1000
time [Myr]

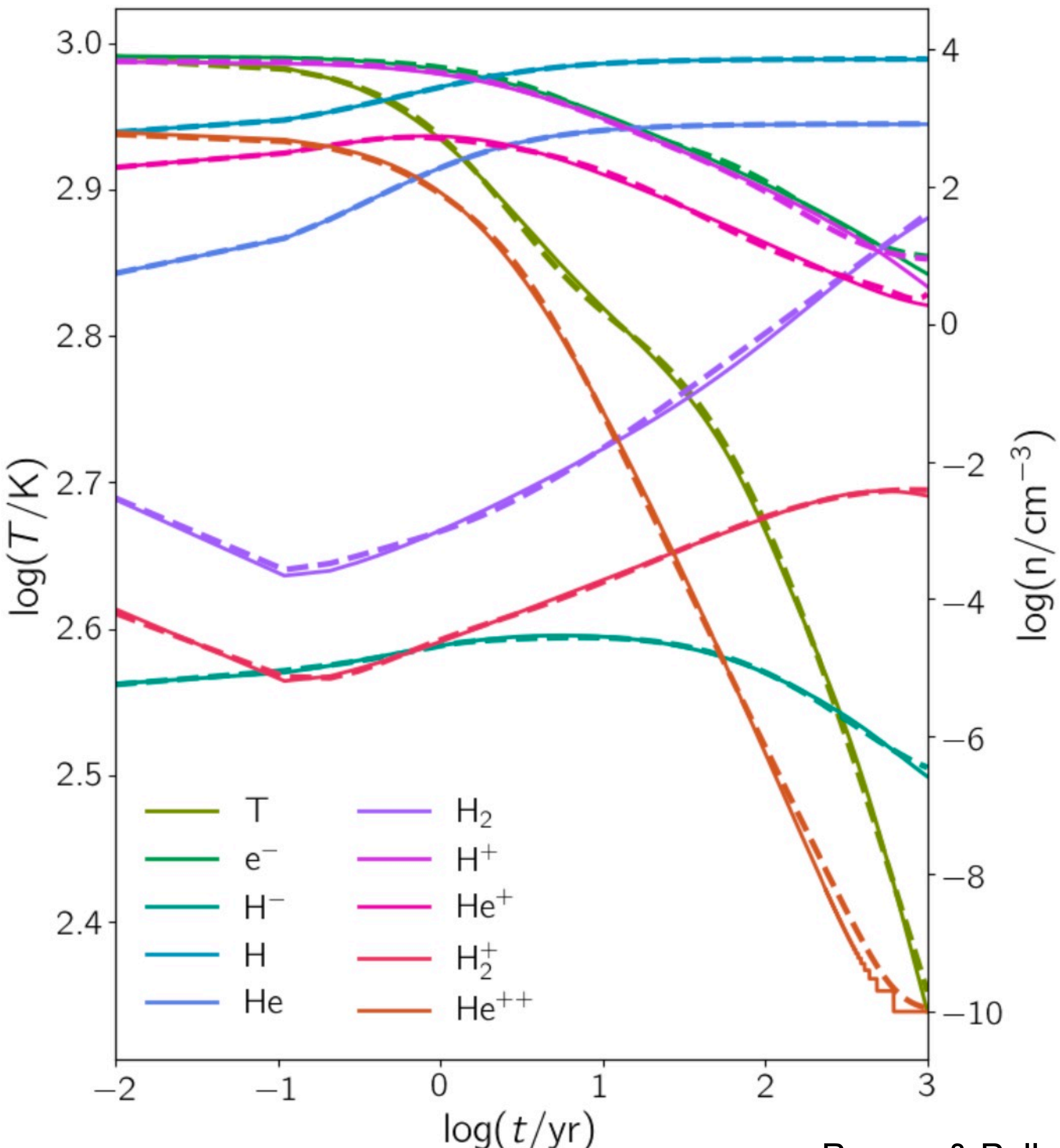
Hu et al. (2019)

- Colman et al: broad agreement between simulations, main disagreement involves mean cloud temperatures
- Hu et al: substantial differences in star formation rate
- Difference in conclusions due to difference in object simulated (massive galaxy vs. dwarf galaxy)?
- Or due to different treatment of SF, feedback?
- Main take away: more work on this urgently needed!

- Running galaxy-scale simulations on 100s or 1000s of CPU is more-or-less routine
- BUT: current Tier 0 machines have 10^4 — 10^5 CPUs
- Unclear how well current approaches scale for such large number of CPUs
- Memory also a **big** issue — lots of physics implies lots of memory required per resolution element
- Increasingly, fastest machines are GPU-based
  - Port everything to GPUs?

AI / Machine learning

- Can machine learning help us?
- In principle, could train a neural net to emulate an entire ISM simulation
- In practice, seems impractical to carry out enough simulations to provide sufficiently rich training data
- Applying the same idea to costly parts of the simulation seems more promising — in particular chemistry



Branca & Pallottini (2024)

Summary

- State-of-the-art simulations of the galactic-scale ISM contain many different physical ingredients
- Many different methods with different strengths, weaknesses — understand the trade-offs when using these simulations (or the data they produce)
- Important goals for the future:
  - More, better comparisons between models
  - How do we deal with a GPU-dominated future?