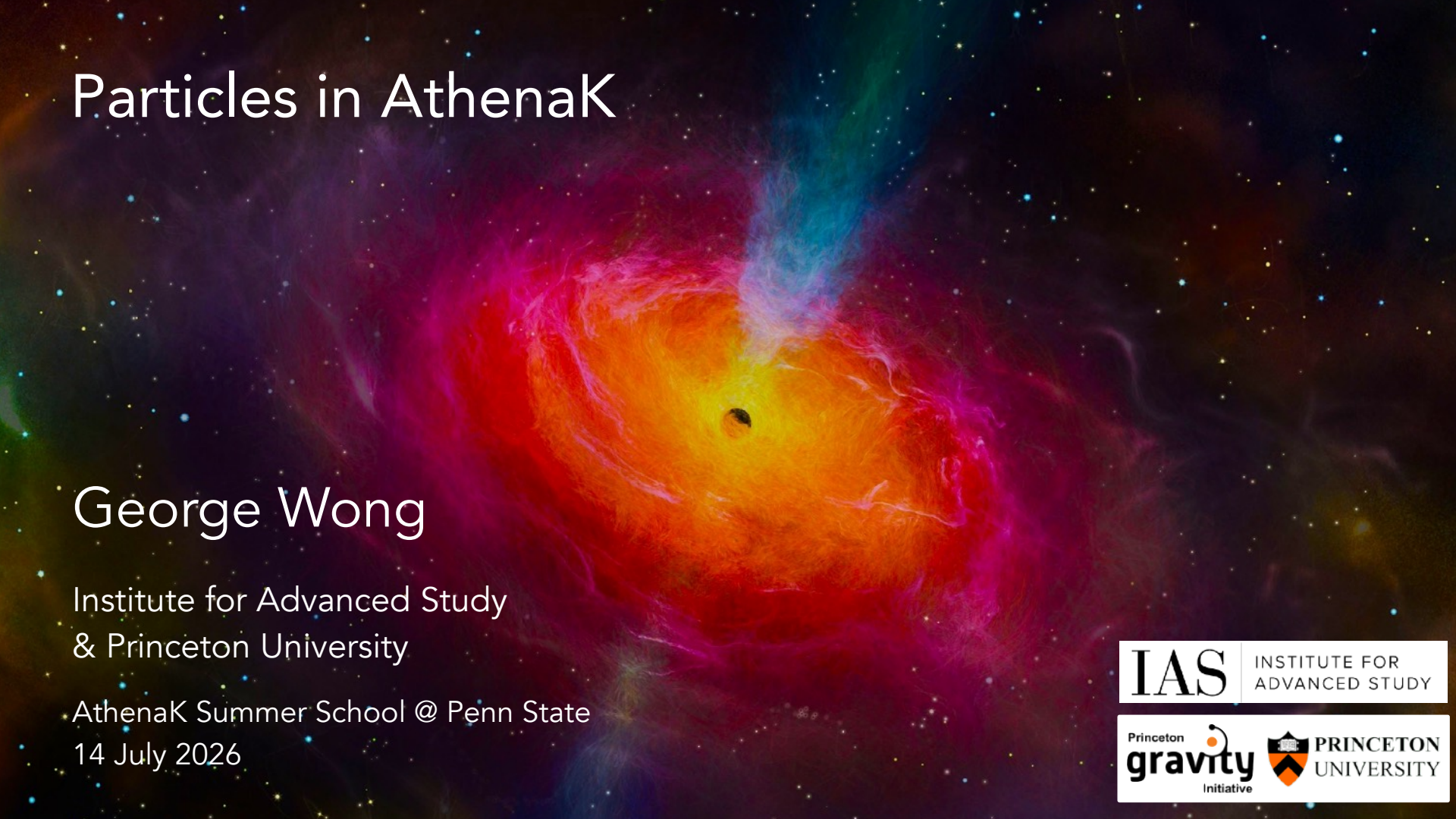
Particles in AthenaK



George Wong

Institute for Advanced Study
& Princeton University

AthenaK Summer School @ Penn State
14 July 2026

Particles in AthenaK

"a practical guide to thinking about, using, and possibly developing particles "

George Wong

Institute for Advanced Study
& Princeton University

AthenaK Summer School @ Penn State
14 July 2026

IAS INSTITUTE FOR ADVANCED STUDY

Princeton gravity Initiative

PRINCETON UNIVERSITY

Particles in AthenaK

"a practical guide to thinking about, using, and possibly developing particles + some applications"

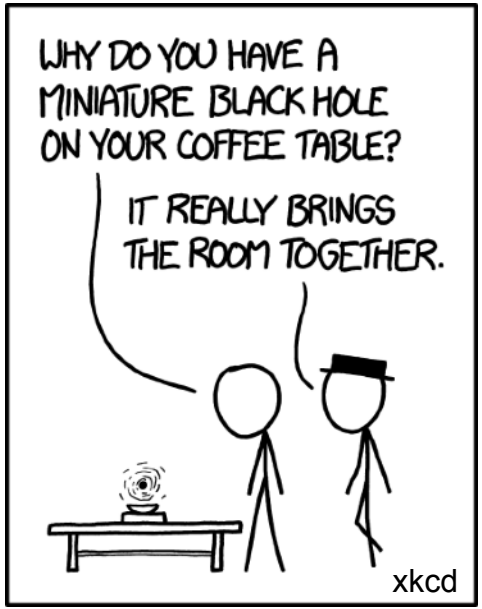
George Wong

Institute for Advanced Study
& Princeton University

AthenaK Summer School @ Penn State
14 July 2026

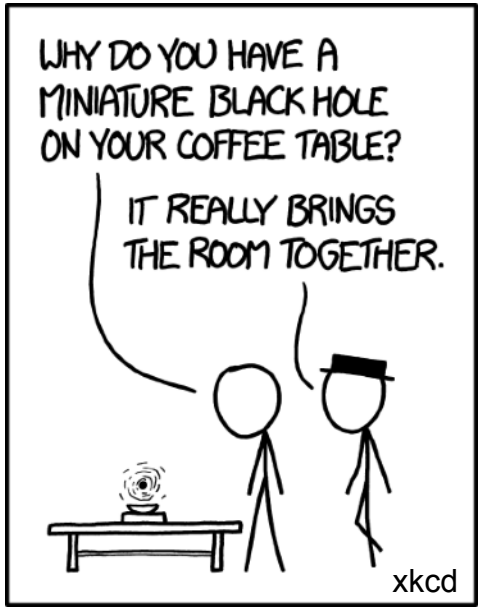
outline

- introduction to particles
  - — what do we mean by particles (taxonomy)?
  - — AthenaK particle overview
  - — nuances / caveats / challenges
- mass transport in non-radiative accretion disks
  - — introduction to these accretion systems
  - — dynamics and kinematics of the flows
  - — transport, mixing, and draining
- populating relativistic jets via disk/jet interactions
  - — why you should care about jets
  - — mass-loading mechanisms
  - — entrainment via the Kelvin-Helmholtz instability



outline

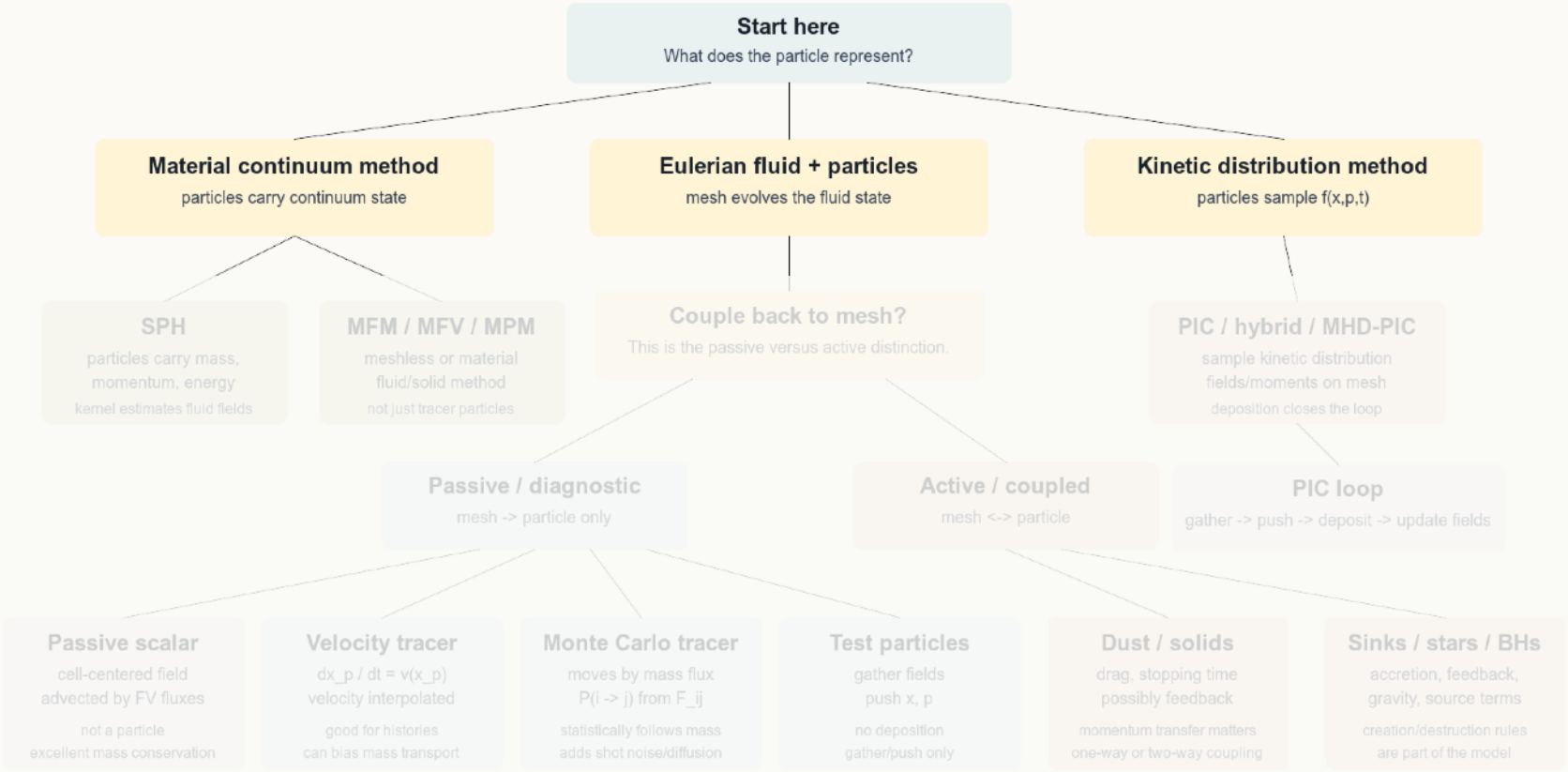
- **introduction to particles**
  - — what do we mean by particles (taxonomy)?
  - — AthenaK particle overview
  - — nuances / caveats / challenges
- mass transport in non-radiative accretion disks
  - — introduction to these accretion systems
  - — dynamics and kinematics of the flows
  - — transport, mixing, and draining
- populating relativistic jets via disk/jet interactions
  - — why you should care about jets
  - — mass-loading mechanisms
  - — entrainment via the Kelvin-Helmholtz instability



… what is a particle?

How to Think About "Particles" in Fluid Codes

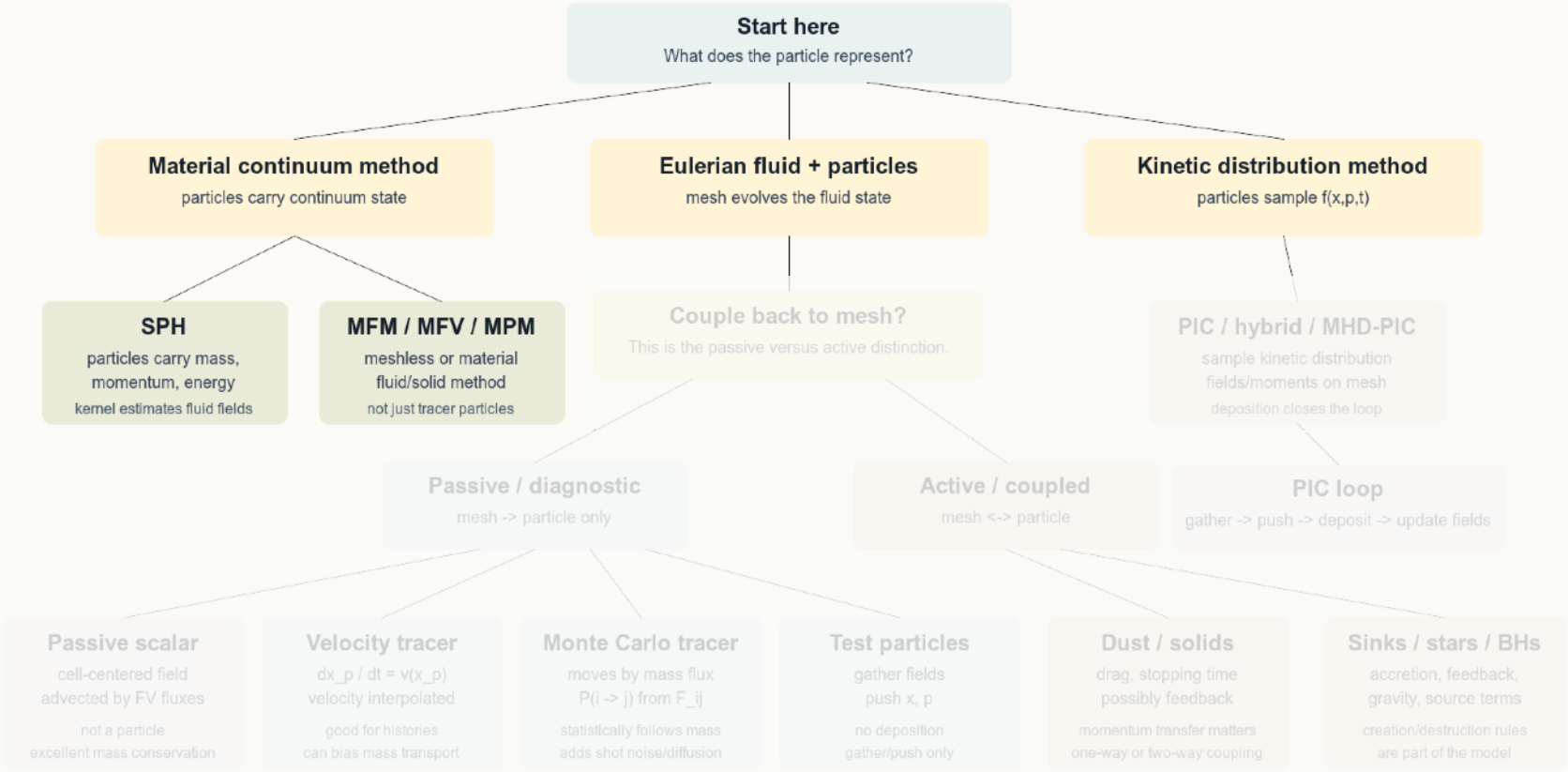
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

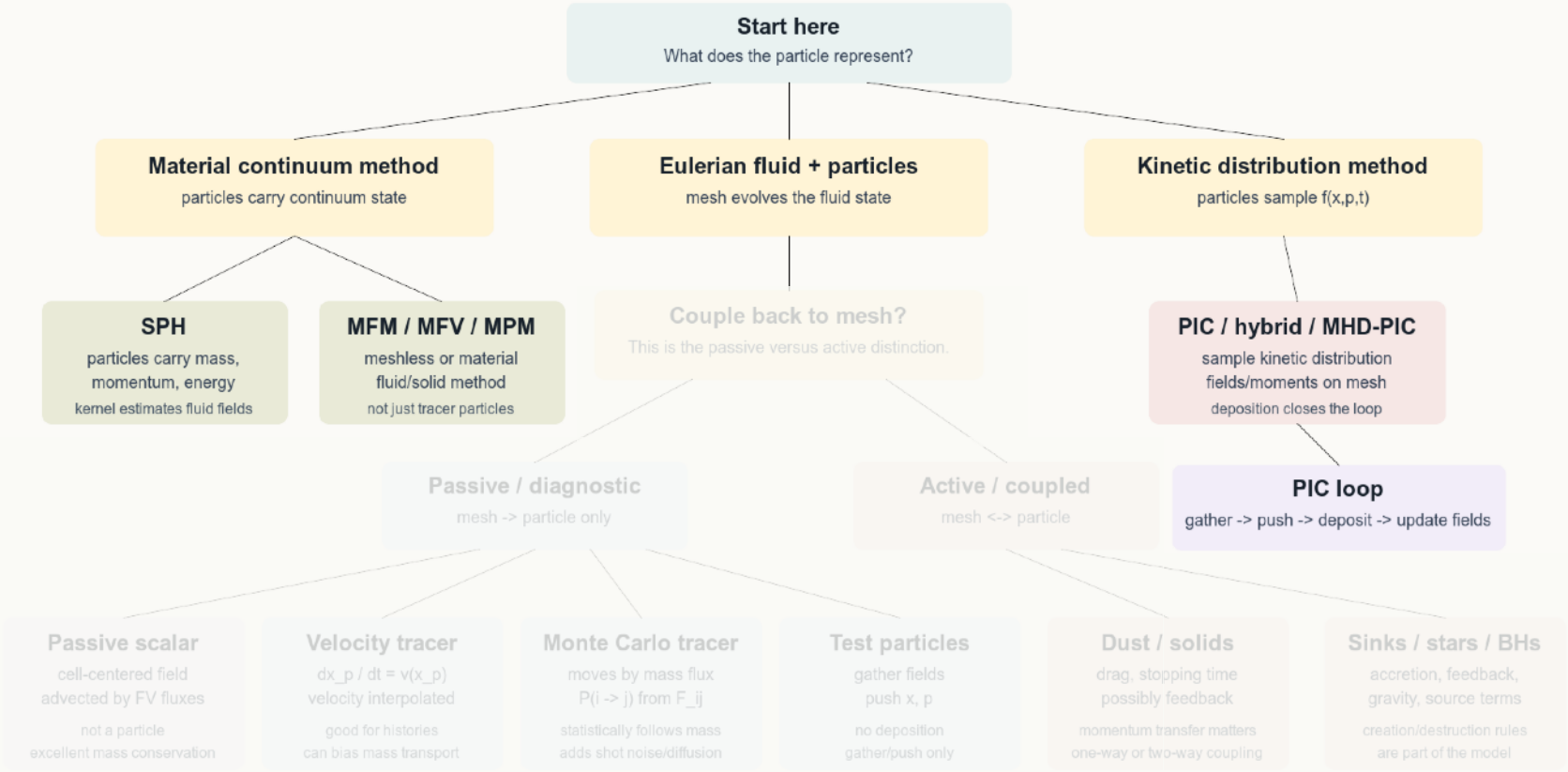
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

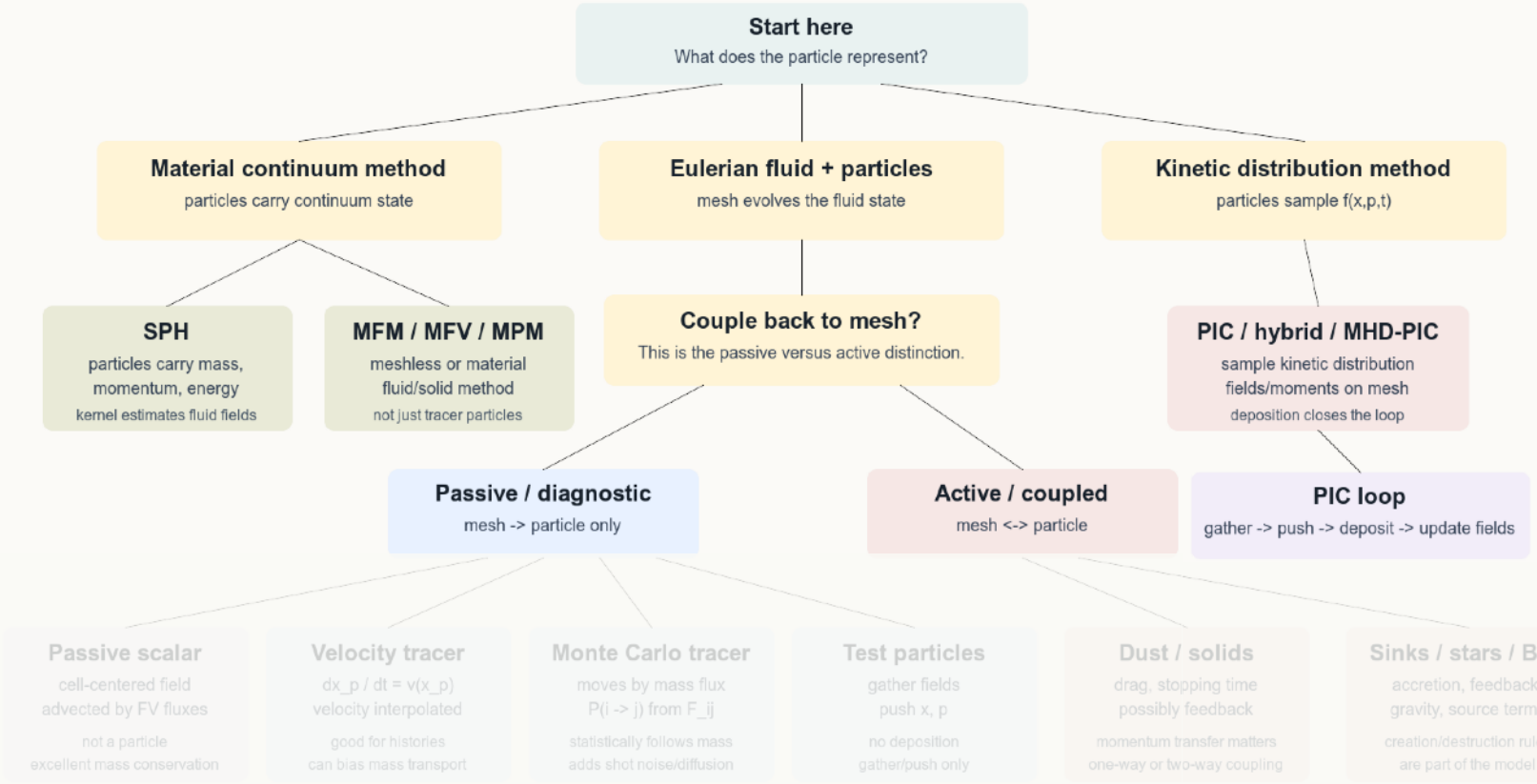
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

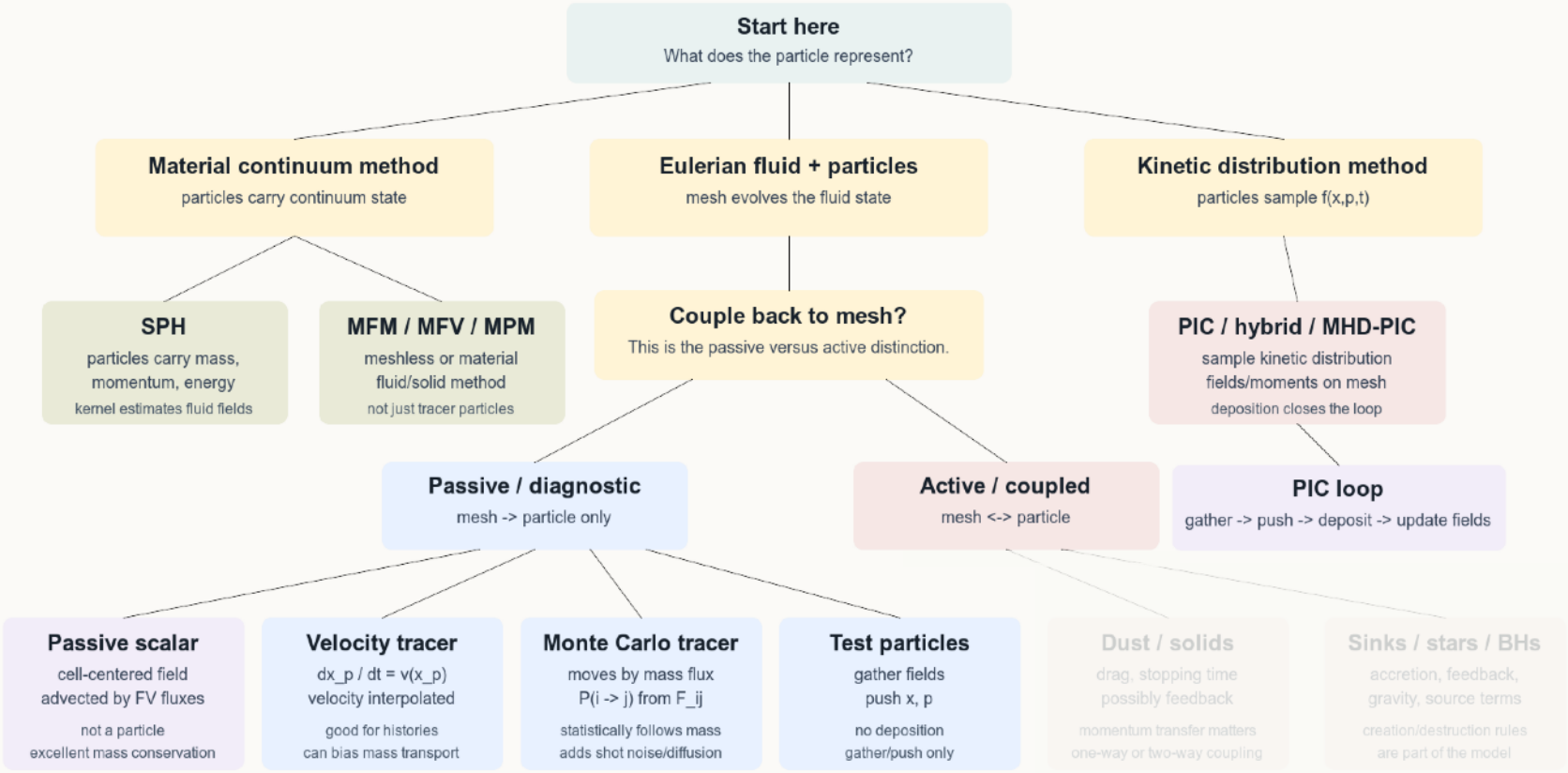
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

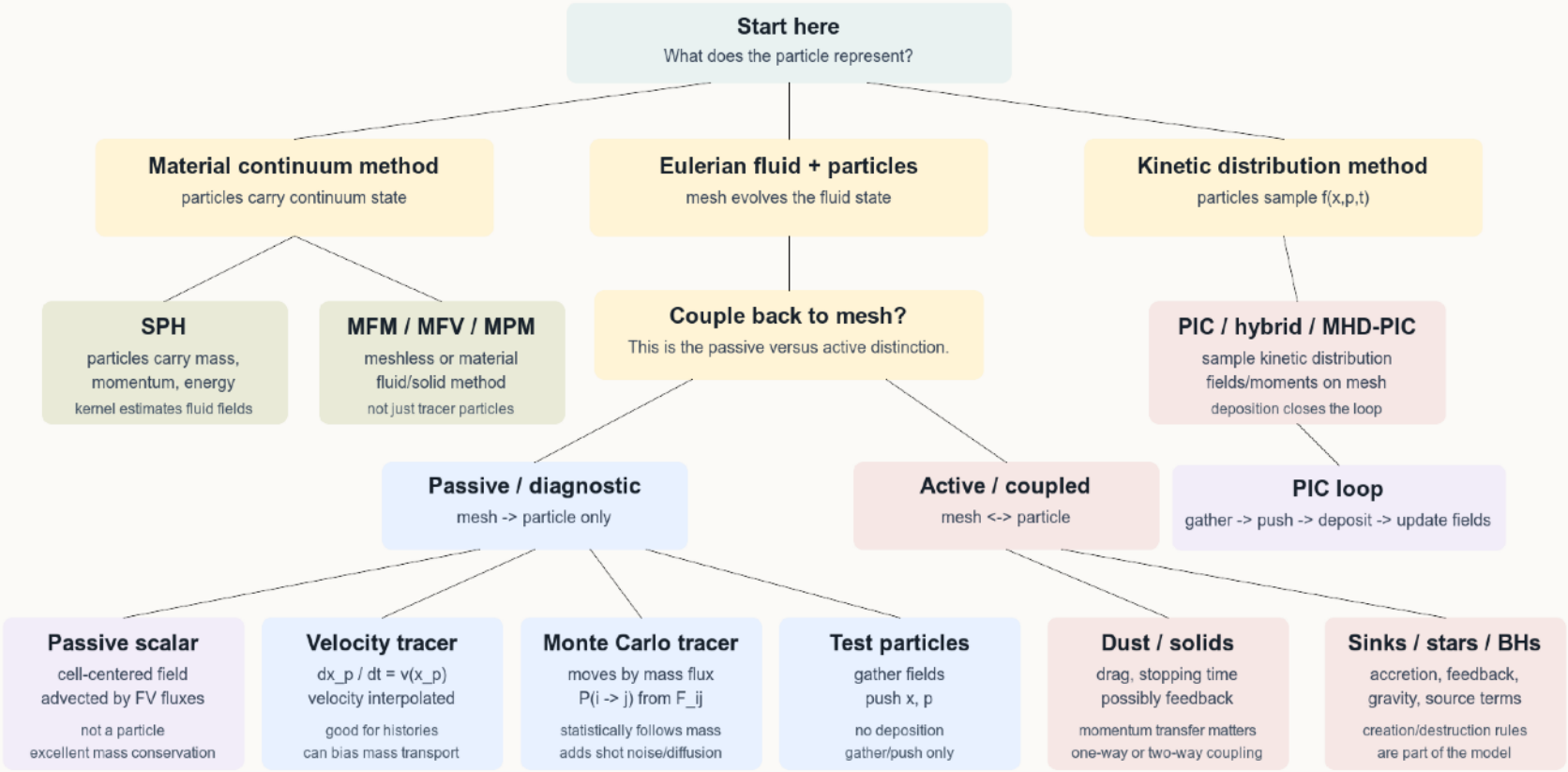
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

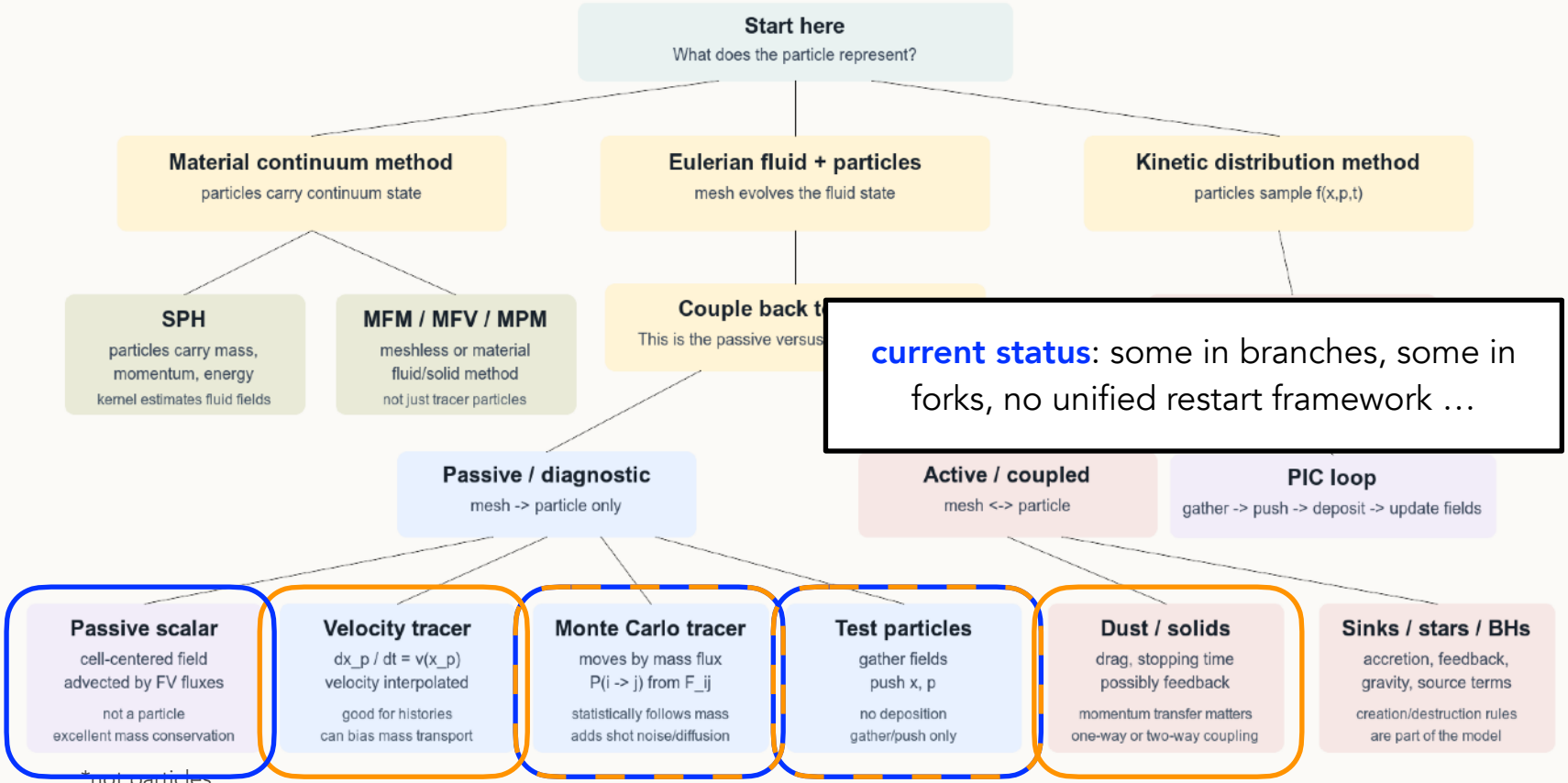
First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

How to Think About "Particles" in Fluid Codes

First ask what the particle represents, how it moves, and whether it changes the mesh state.



*not particles

general particle data structure

```
class Particles {
  friend class ParticlesBoundaryValues;
 public:
  Particles(MeshBlockPack *ppack, ParameterInput *pin);
  ~Particles();

  // data
  ParticleType particle_type;
  int nprtcl_thispack;             // number of particles this MeshBlockPack
  int nrdata, nidata;
  DvceArray2D<Real> prtcl_rdata;   // real number properties each particle (x,v,etc.)
  DvceArray2D<int>  prtcl_idata;   // integer properties each particle (gid, tag, etc.)
  Real dtnew;

  Real min_radius;  // particles that enter within this radius will not be updated

  ParticlesPusher pusher;

  // Boundary communication buffers and functions for particles
  ParticlesBoundaryValues *pbval_part;

  // container to hold names of TaskIDs
  ParticlesTaskIDs id;

  // functions
  ...
```

general (passive) particle update procedure

```
for each timestep n -> n+1:

  set dt from fluid / driver

  if particles are scheduled before fluid integrator:
      EvolvePassiveParticles(dt)

  advance fluid/MHD/GR fields through normal AthenaK time integrator

  if particles are scheduled after fluid integrator:
      EvolvePassiveParticles(dt)

  write outputs if needed
```

general (passive) particle update procedure

```
for each timestep n -> n+1:

  set dt from fluid / driver

  if particles are scheduled bef
      EvolvePassiveParticles(dt)

  advance fluid/MHD/GR fields th

  if particles are scheduled aft
      EvolvePassiveParticles(dt)

  write outputs if needed
```

```
EvolvePassiveParticles(dt):

  Task 1: Push
      for each local particle p:
          > load particle state
          > use Particles::Push(ParticlesPusher) to update state
          > store particle state

  Task 2: NewGID / ownership update
      for each particle:
          determine new(?) MeshBlock, consider boundary conditions
          mark particles that left the local rank
          freeze or destroy
  ...
  Task 5: SendP
      pack outgoing particles + send to new ranks

  Task 6: RecvP
      receive incoming particles
      insert into local particle arrays
      fill holes left by sent/destroyed particles
      resize arrays if needed
  ...
```

general (passive) particle update procedure

```
for each timestep n -> n+1:

  set dt from fluid / driver

  if particles are scheduled bef
      EvolvePassiveParticles(dt)

  advance fluid/MHD/GR fields th

  if particles are scheduled aft
      EvolvePassiveParticles(dt)

  write outputs if needed
```

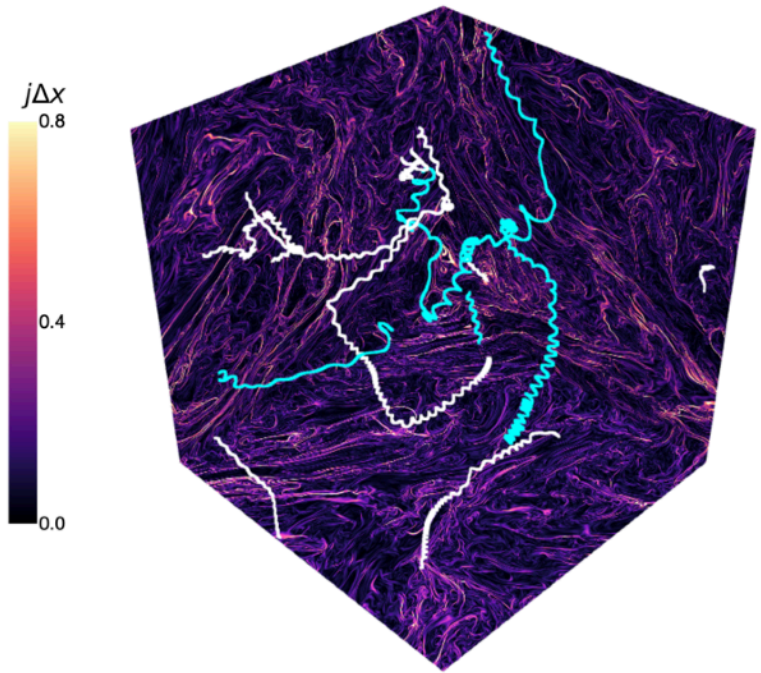
```
EvolvePassiveParticles(dt):

  Task 1: Push
      for each local particle p:
          > load particle state
          > use Particles::Push(ParticlesPusher) to update state
          > store particle state

  Ta
  ..
  Ta
  Ta
  ..
```

```
Particles::Push(ParticlesPusher)

          if pusher == drift:
              x_p <- x_p + velocity_p * dt_eff

          if pusher == boris / GR:
              gather mesh fields at particle position
                  B, E, metric data
              update momentum/velocity with Boris / GR pusher
              update position

          if pusher == lagrangian_mc:
              read local density u1(IDN)
              read saved density fluxes uflxidnsaved
              compute outward flux probabilities
              draw random number
              move particle by at most one cell
```

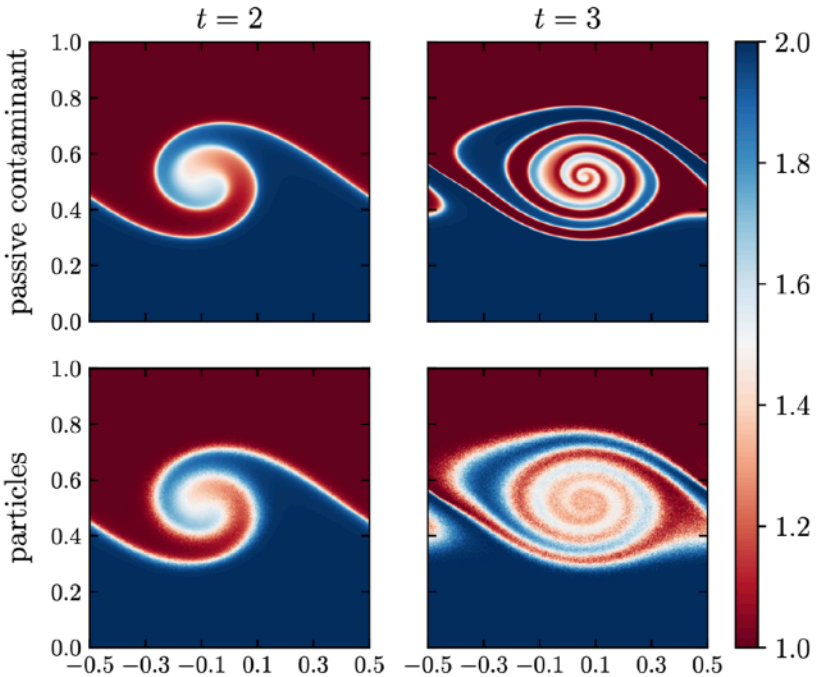
particle types that have been implemented

Boris pusher (e.g., charged particles)

```
BorisStep(p, dt, update_position):

  x, u <- particle position and momentum

  x_half <- drift x by dt/2 using current u

  E, B, metric <- gather at x_half
  u <- transform to local pusher frame

  u <- u + (q/m) E dt/2          # electric half-kick
  u <- rotate u about B by dt    # magnetic Boris rotation
  u <- u + (q/m) E dt/2          # electric half-kick

  u <- transform/store updated momentum

  if update_position:
      x <- drift x_half by dt/2 using updated u
      store x


FullGRStep(p, dt):   # operator split for motion in curved space

  BorisStep(p, dt/2, update_position=false)

  GeodesicStep(p, dt)

  BorisStep(p, dt/2, update_position=false)
```



particle types that have been implemented



Monte Carlo Lagrangian tracers

```
LagrangianMCStep(p):

  cell <- cell containing particle

  read rho in cell
  read saved density fluxes through cell faces

  for each face:
      P(face) <- max(outward flux through face, 0) / cell_mass

  draw random number r

  if r selects a face:
      move particle to neighboring cell across that face
  else:
      leave particle in current cell

  store last move/refinement info
```

general (~~passive~~ ***active**) particle update procedure

when particles are active, the problem is more complicated:

momentum conservation
=> dp(particle) + dp(fluid) = 0

energy conservation
=> drag + heating = 0

stiffness (cf. Jake Simon's talks)

```
for each timestep n -> n+1:

  set dt from fluid / driver

  if particles are scheduled before fluid integrator:
      EvolvePassiveParticles(dt)

  advance fluid/MHD/GR fields through normal AthenaK time integrator

  if particles are scheduled after fluid integrator:
      EvolvePassiveParticles(dt)

  write outputs if needed
```

need: conservation laws, particle -> grid kernel, implicit solve, …

general (~~passive~~ ***active**) particle update procedure

when particles are active, the problem is more complicated:

momentum conservation
=> dp(particle) + dp(fluid) = 0

energy conservation
=> drag + heating = 0

stiffness (cf. Jake Simon's talks)

```
for each active particle:
    define 27-cell deposition cloud around particle
    compute weights w_i
    distribute weighted particle momentum into those cells

solve (local) implicit momentum exchange:
    particle momentum loss/gain depends on updated gas velocity
    gas momentum gain/loss depends on updated particle momentum

for each active particle:
    update particle momentum using solved impulse J_p

for each affected gas cell:
    deposit opposite impulse: Delta P_i = - sum_p w_i J_p

if conserving total energy:
    update gas energy / heat with consistent work term
```

need: conservation laws, particle -> grid kernel, implicit solve, …

practical thoughts and caveats

Before You Run Particles

Define the **particle**:
tracer, charged test particle, MC mass tracer, dust, sink?

Consider the **state**:
position, momentum/velocity, weight, tag, owner block

Check the **coupling**:
passive gather only, or active deposition back to mesh?

Describe the **lifecycle**:
initialization, injection, destruction, boundary behavior

Select how to **output**:
output cadence and format .. beware disk cost!

Choose how to perform **validation**:
what statistic should converge?
orbit error, mass distribution, flux statistics, restart continuity

Be Warned / Likely To Change

Particle behavior/implementation/support is **branch-dependent**.

Output exists, but **restarts not (formally) implemented** yet.

Current implementations** of particles are **passive**:
no back reaction … but:

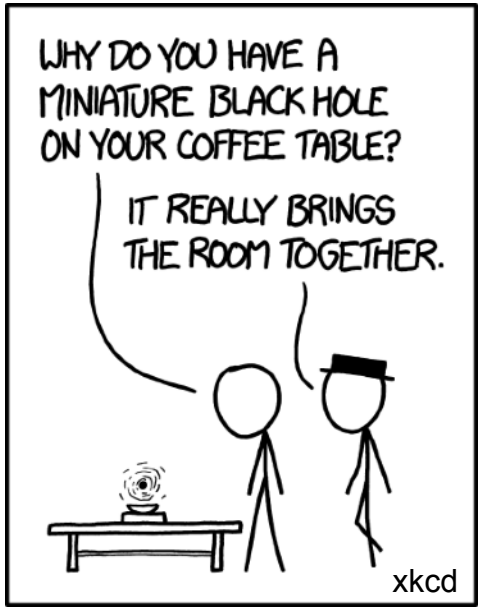
Active particles are a larger, ongoing project:
coupling, deposition, conservation, and implicit stiff source terms
need to be address conceptually.

AMR behavior still **needs explicit validation**, especially across boundaries.

Particle identity can be subtle: IDs, ownership, deletion/freezing, and communication all matter.

outline

- introduction to particles
  - what do we mean by particles (taxonomy)?
  - AthenaK particle overview
  - nuances / caveats / challenges
- **mass transport in non-radiative accretion disks**
  - introduction to these accretion systems
  - dynamics and kinematics of the flows
  - transport, mixing, and draining
- populating relativistic jets via disk/jet interactions
  - why you should care about jets
  - mass-loading mechanisms
  - entrainment via the Kelvin-Helmholtz instability



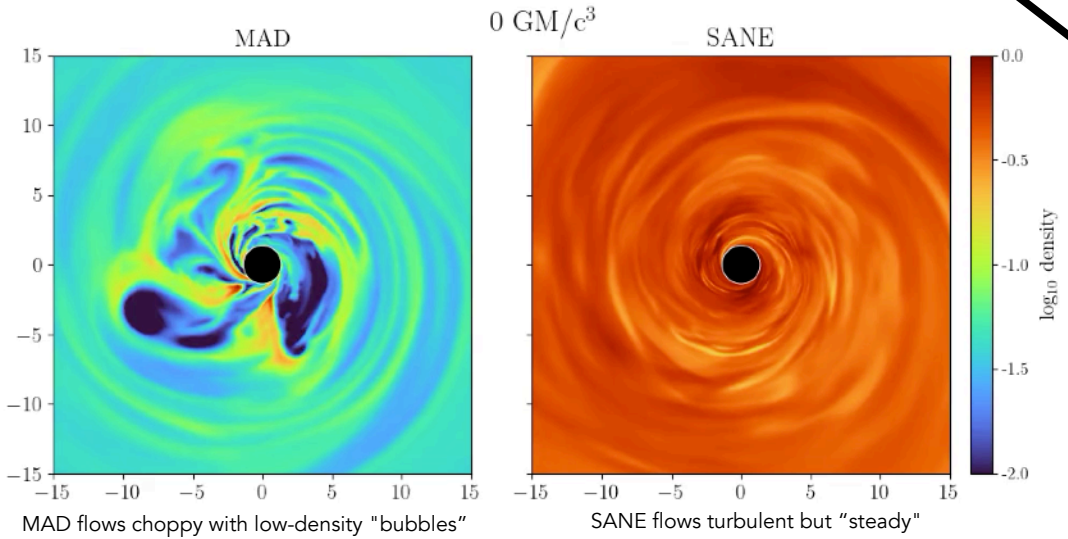
application:
kinematics + dynamics of thick-disk accretion

magnetic fields in black hole accretion

magnetic fields may **follow the turbulence** (SANE)

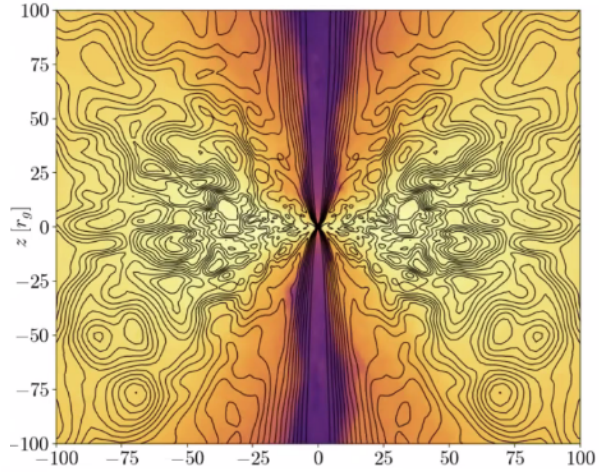
or be **dynamically important** (MAD)

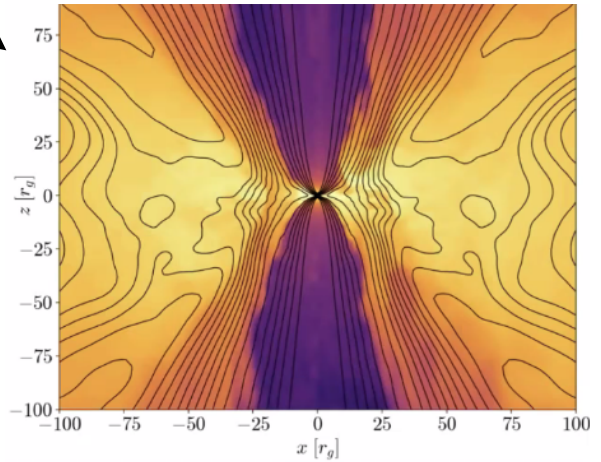
* midplane slice through simulation



MAD flows choppy with low-density "bubbles"

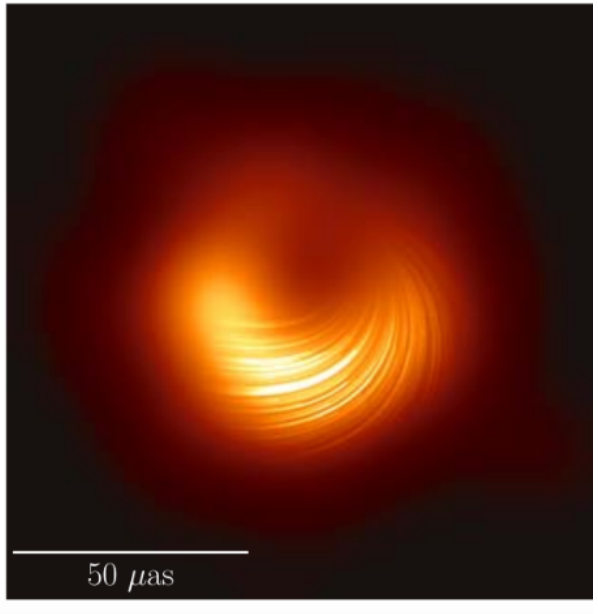
SANE flows turbulent but "steady"

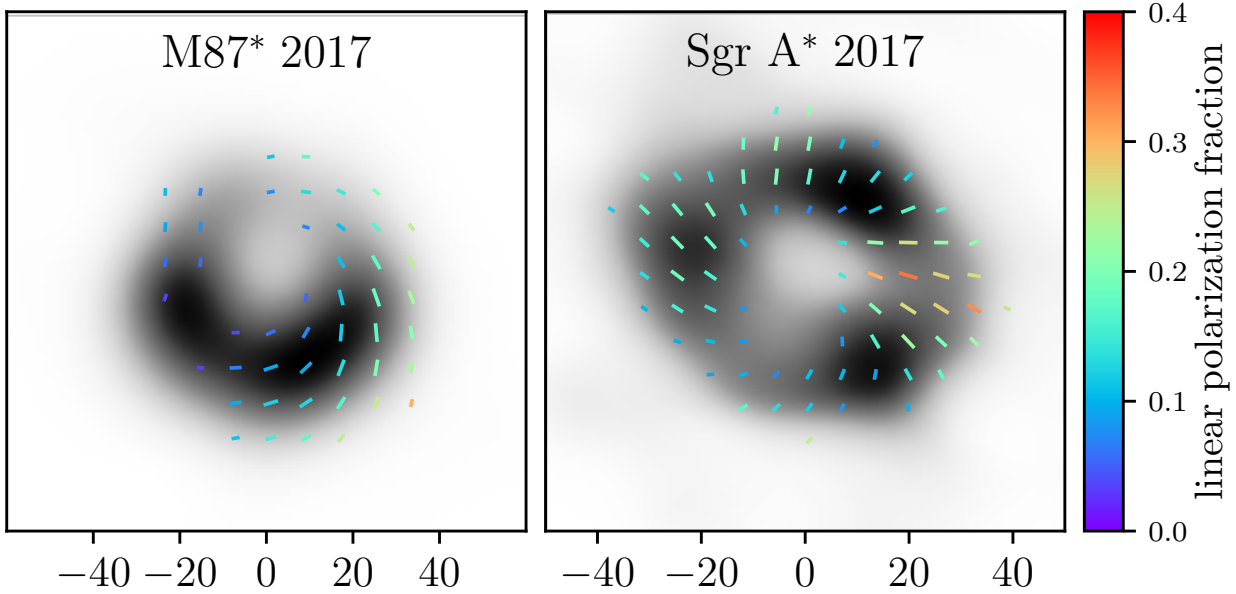




Wong+ 2022

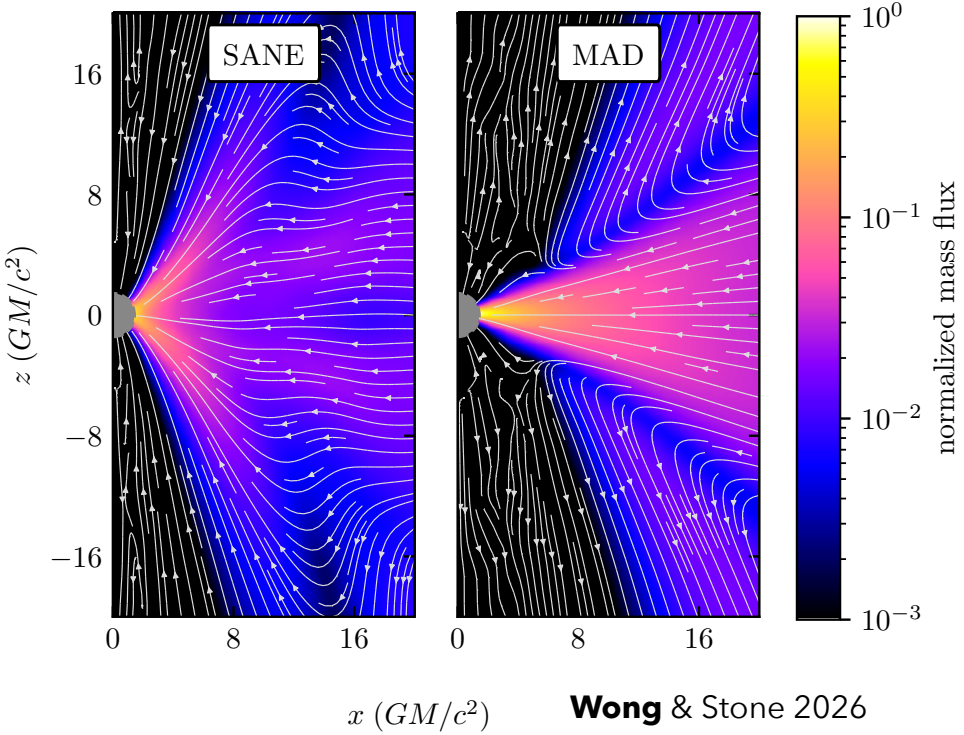
why do we care about the magnetic fields?





EHTC (2019, 2021, 2024)

SANE and MAD accrete by different channels

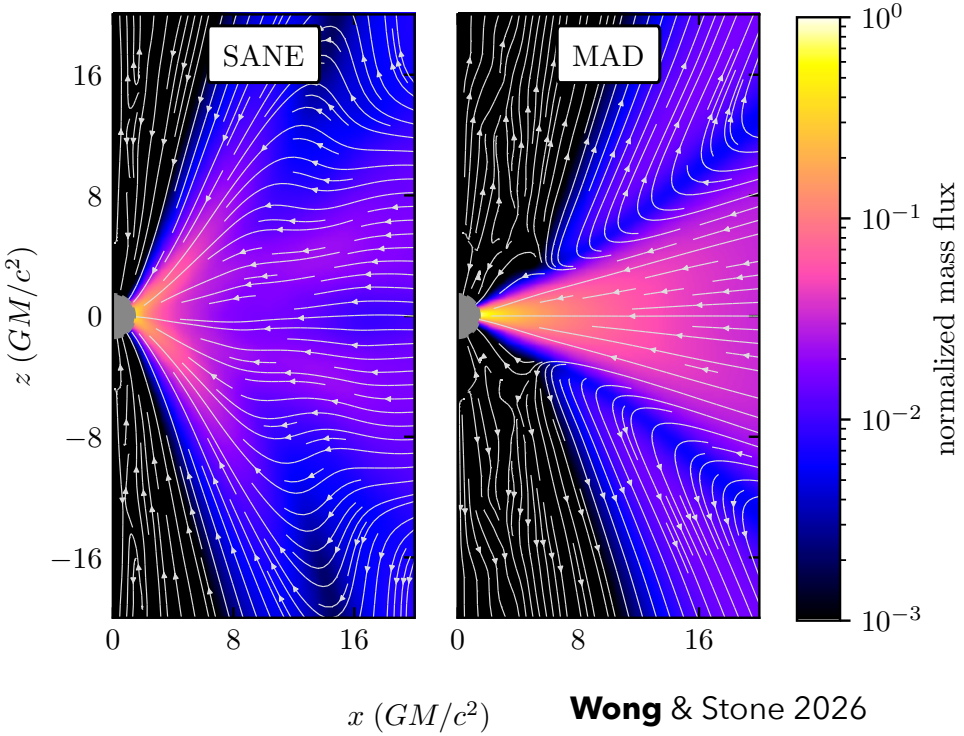


Wong & Stone 2026

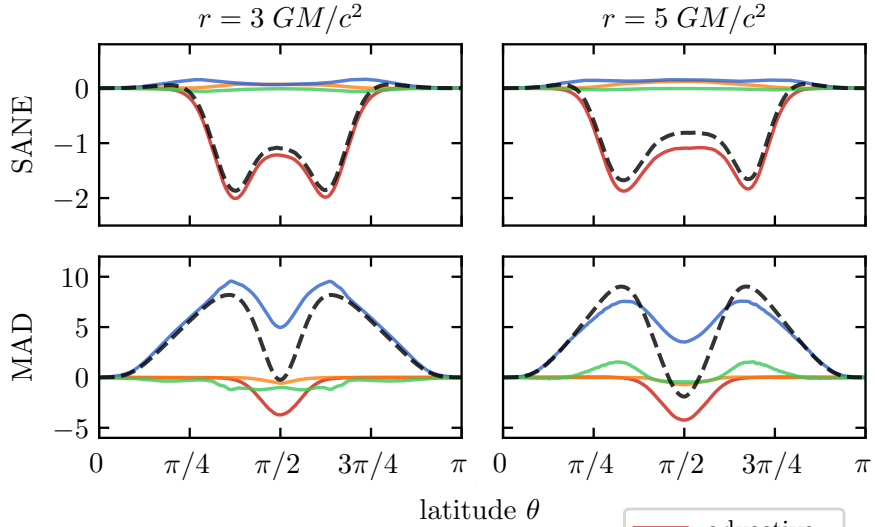
In SANEs, **vertical stratification** allows more efficient Maxwell angular momentum transport where plasma-β smaller

In MADs, nearly all of the disk is low β and **accretion can proceed** at all latitudes

SANE and MAD accrete by different channels



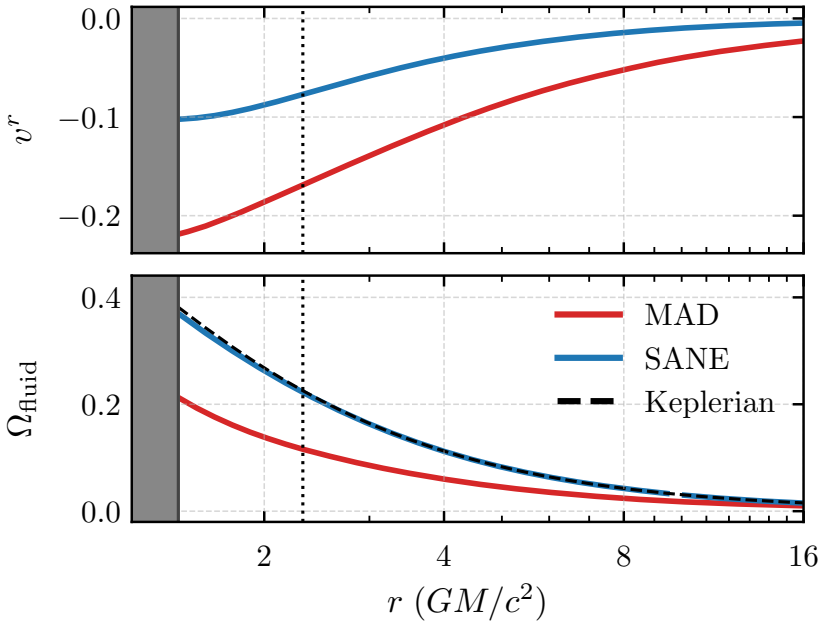
Wong & Stone 2026

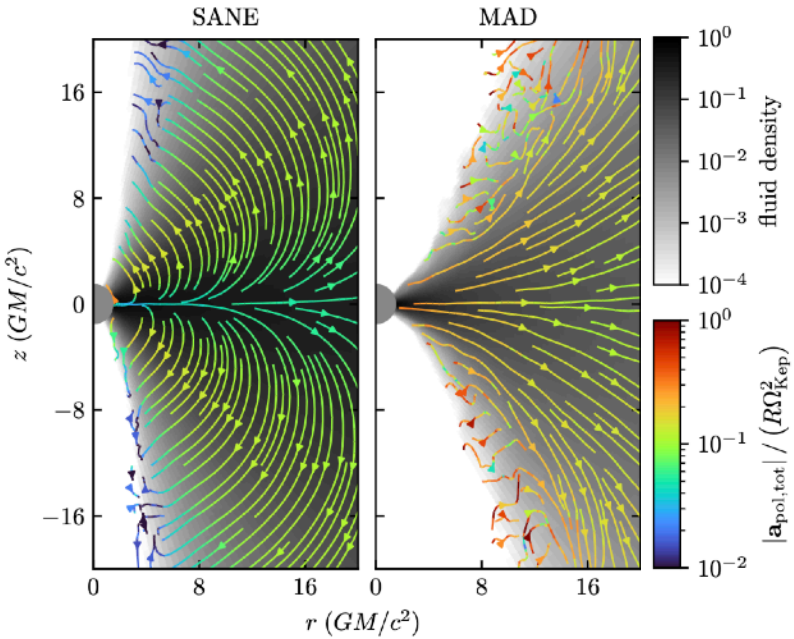


SANE dominated by **advective transport** with **Reynolds** and **Maxwell** in disk/surface

MAD dominated by **Maxwell stresses**

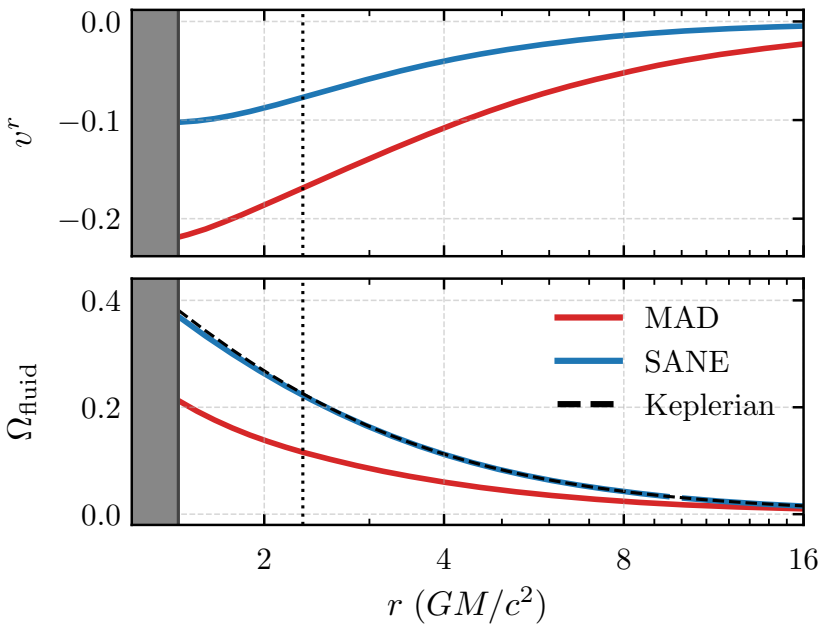
SANE and MAD have qualitatively distinct kinematics

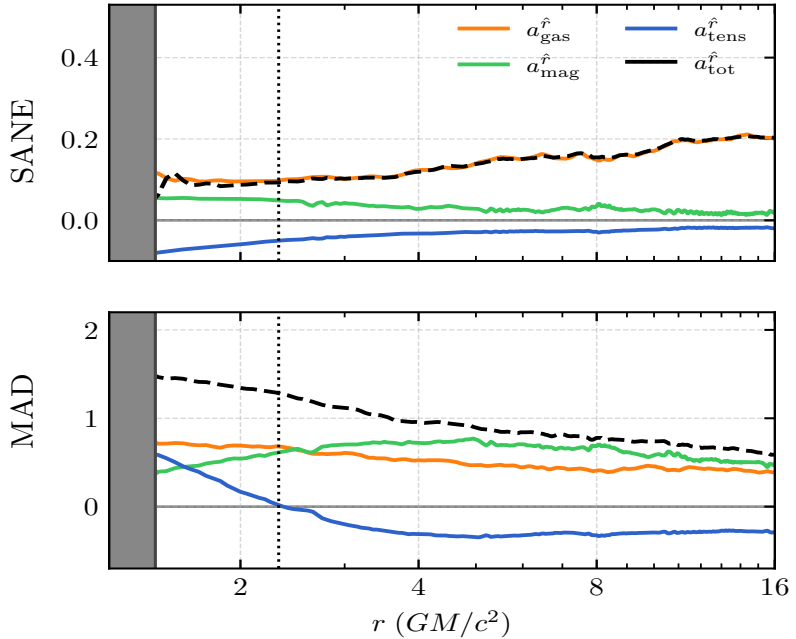




Wong & Stone 2026

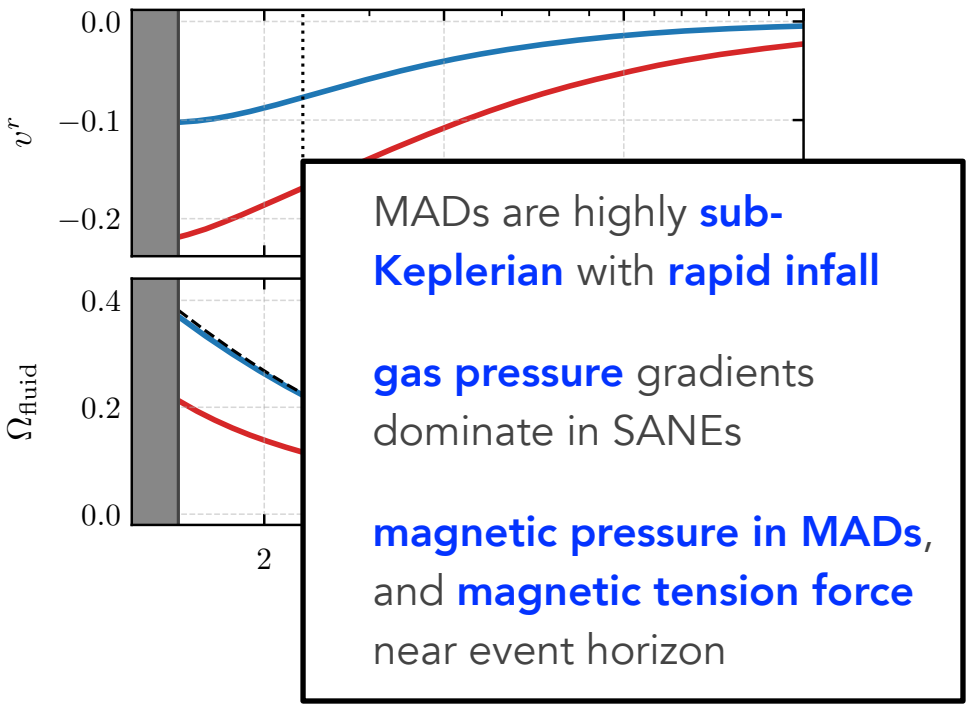
SANE and MAD have qualitatively distinct kinematics





Wong & Stone 2026

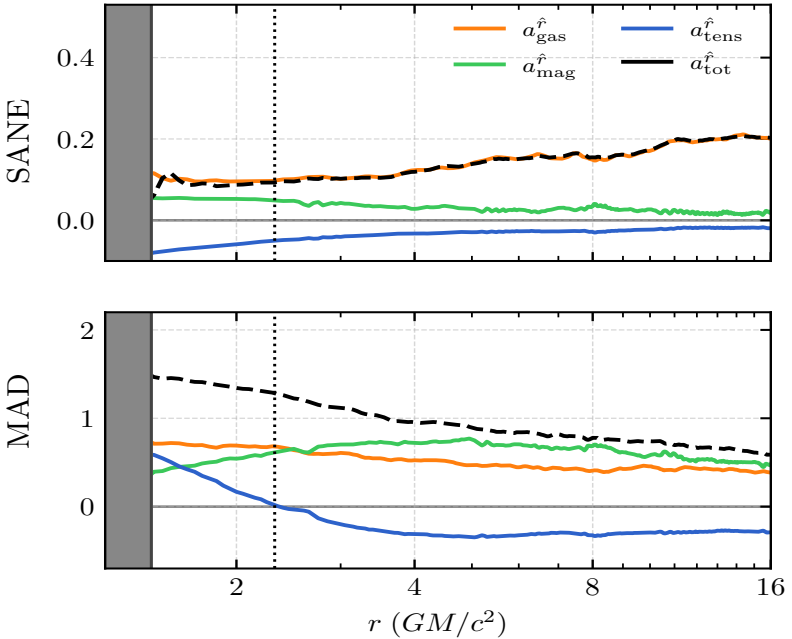
SANE and MAD have qualitatively distinct kinematics



MADs are highly **sub-Keplerian** with **rapid infall**

gas pressure gradients dominate in SANEs

magnetic pressure in MADs, and **magnetic tension force** near event horizon

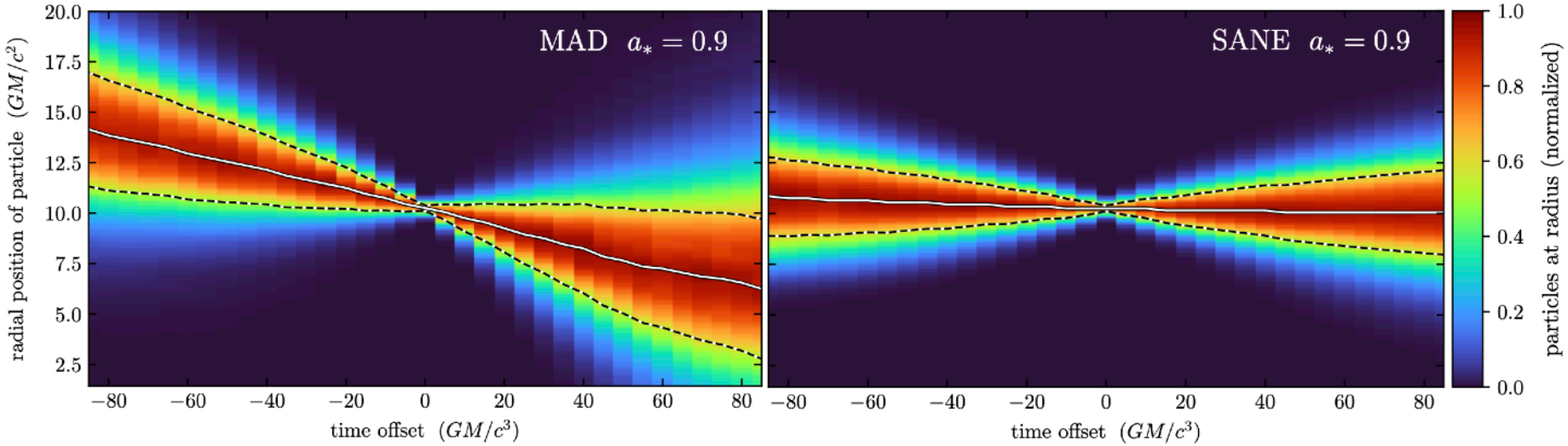


Wong & Stone 2026

mixing and advection

mass flux generally governed by advection and mixing (not necessarily diffusive … K not constant):

$$\partial_t C = \partial_r \left(K \, \partial_r C - v \, C \right) + s$$



Wong, Medeiros, Stone 2025

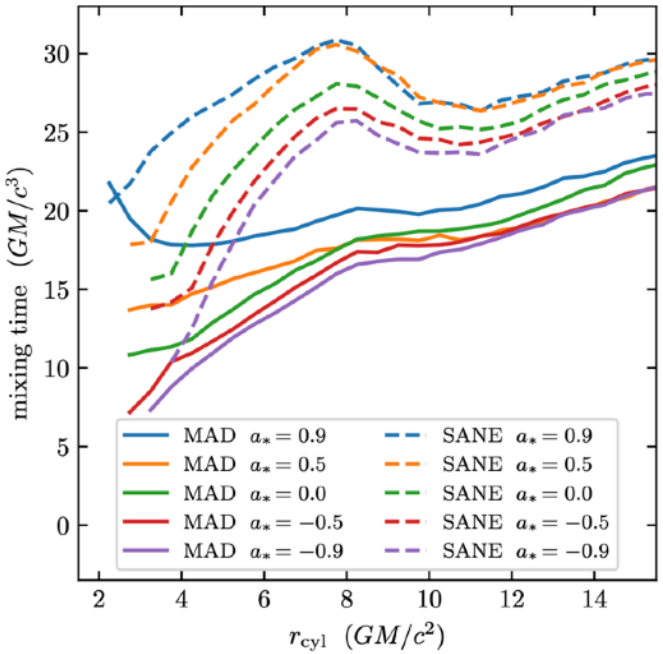
mixing and advection

Mixing consistently **faster in MADs** vs. SANEs.

Retrograde flows mix faster than prograde.

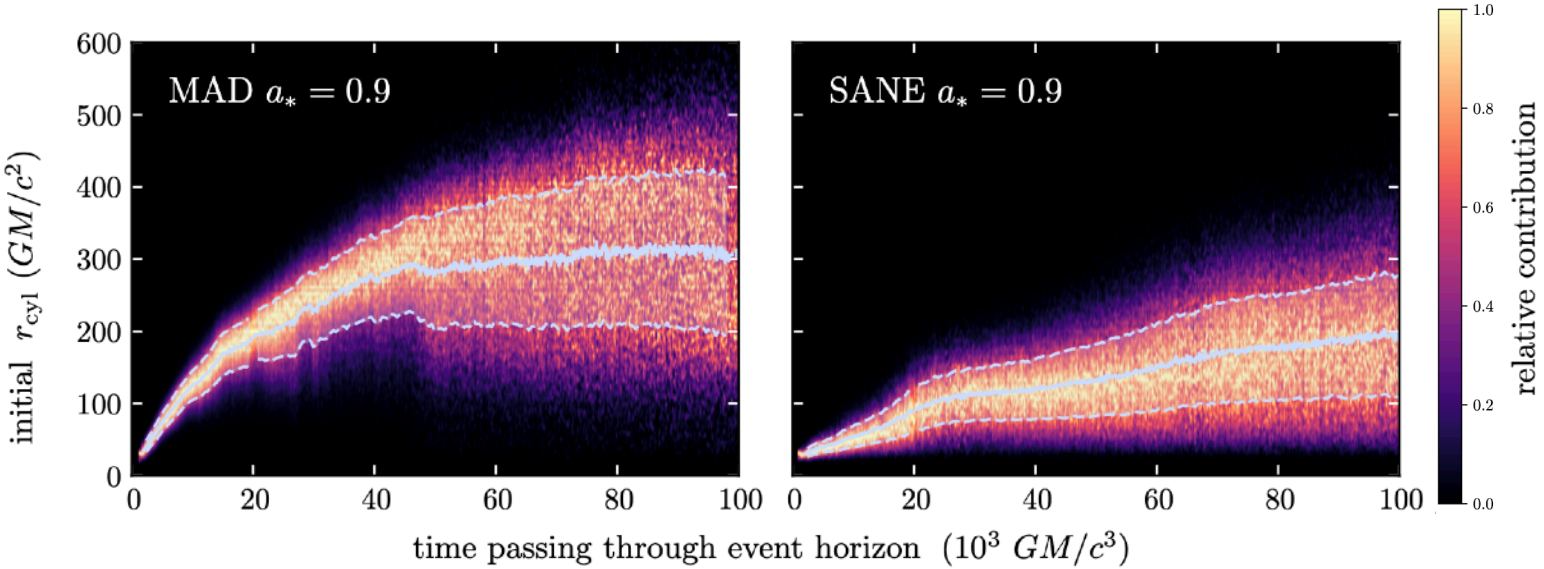
Mixing times generally **decrease toward hole** …

but not monotonic and may reflect the plasma state being "frozen" at larger radii before plunging toward the black hole.



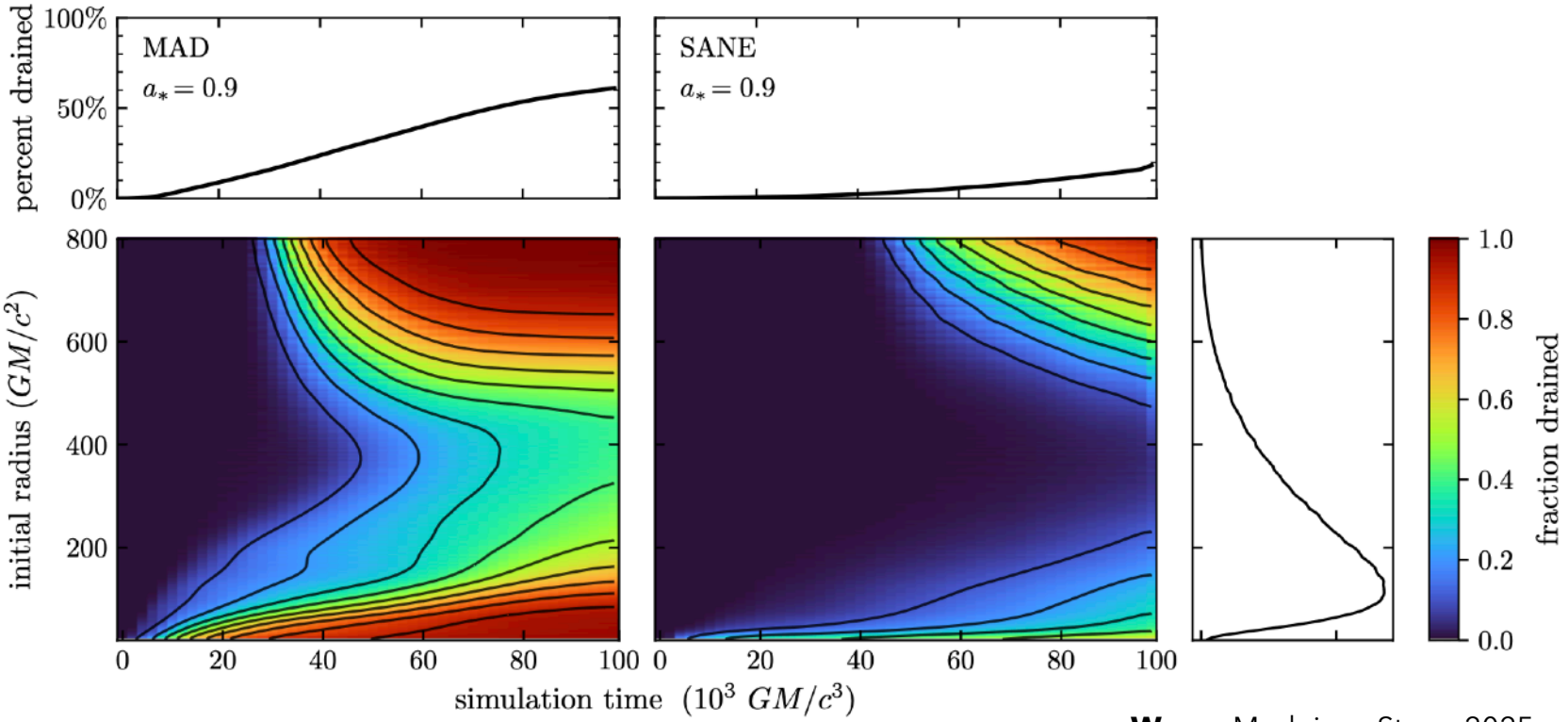
Wong, Medeiros, Stone 2025

disk draining



Wong, Medeiros, Stone 2025

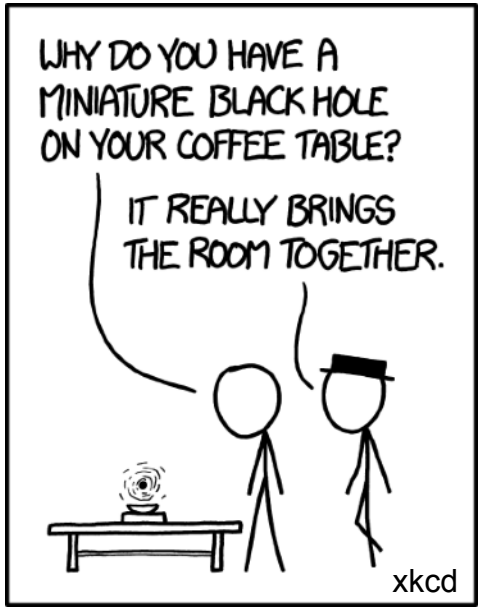
disk draining



Wong, Medeiros, Stone 2025

outline

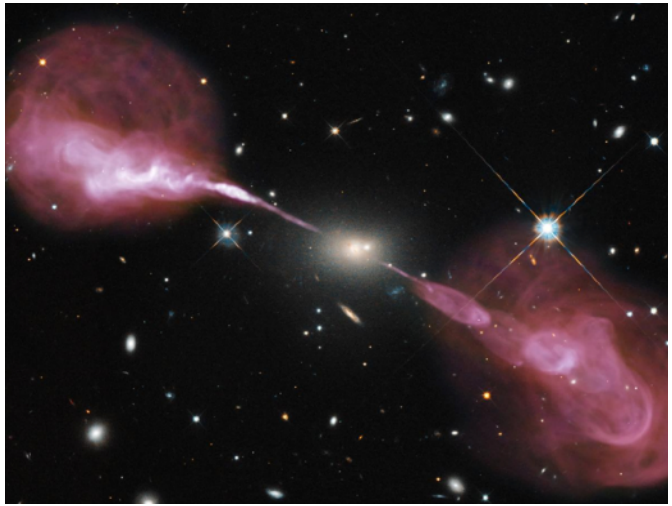
- introduction to particles
  - — what do we mean by particles (taxonomy)?
  - — AthenaK particle overview
  - — nuances / caveats / challenges
- mass transport in non-radiative accretion disks
  - — introduction to these accretion systems
  - — dynamics and kinematics of the flows
  - — transport, mixing, and draining
- **populating relativistic jets via disk/jet interactions**
  - — why you should care about jets
  - — mass-loading mechanisms
  - — entrainment via the Kelvin-Helmholtz instability



application:
understanding mass-loading in relativistic jets

relativistic jets launched by black holes

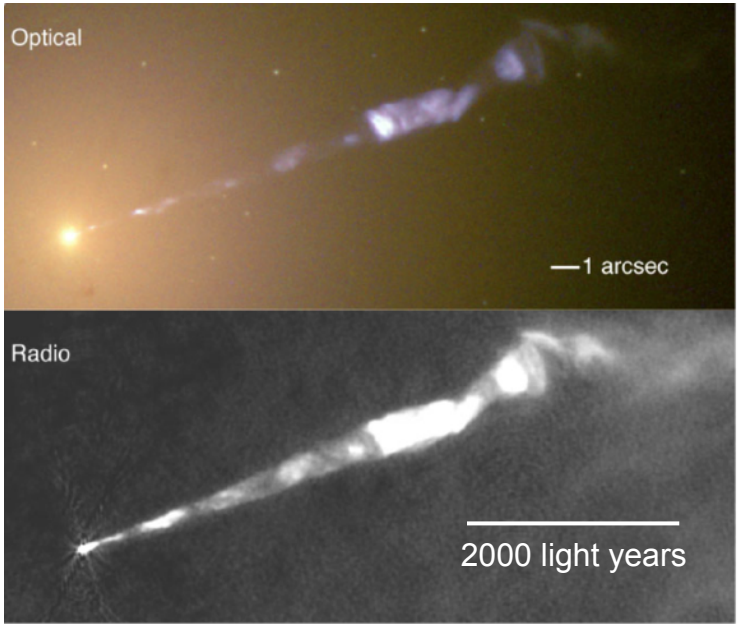
Hercules A



NASA, ESA, Baum & O'Dea, Perley & Cotton, Hubble Heritage

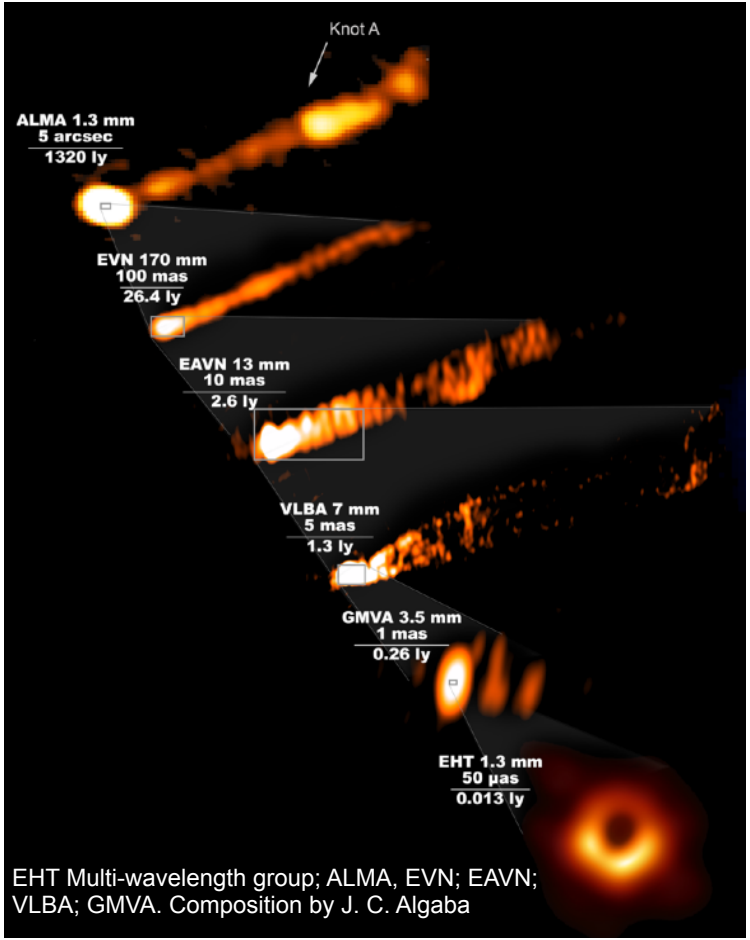
Centaurus A

NASA/CXC/CfA/R.Kraft et al./MPIfR/ESO/WFI/APEX/A.Weiss et al.

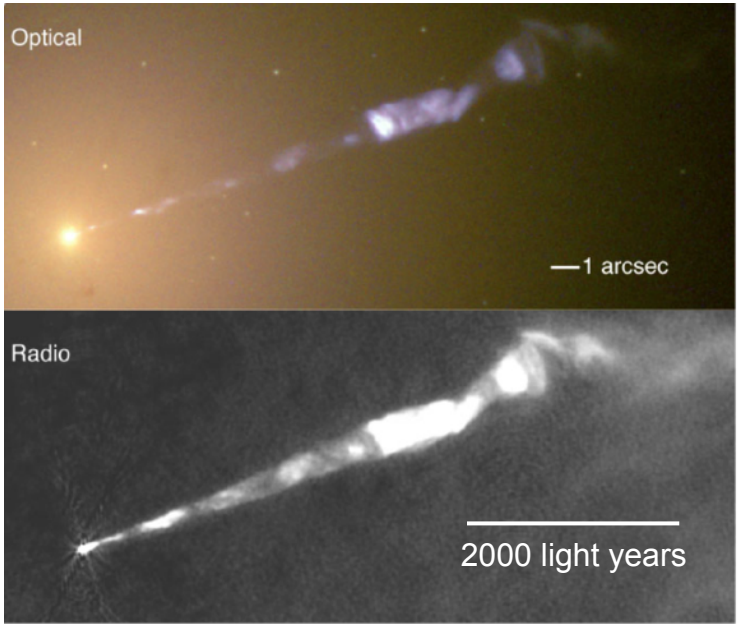


credit: J. Biretta

relativistic jets launched by black holes

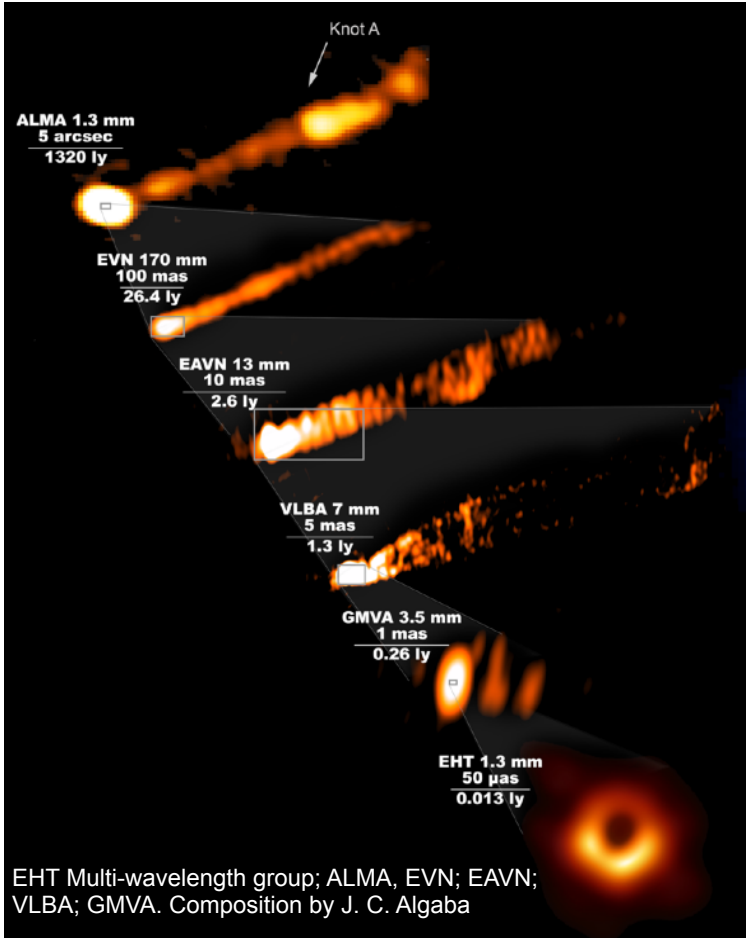


EHT Multi-wavelength group; ALMA, EVN; EAVN; VLBA; GMVA. Composition by J. C. Algaba



credit: J. Biretta

relativistic jets launched by black holes

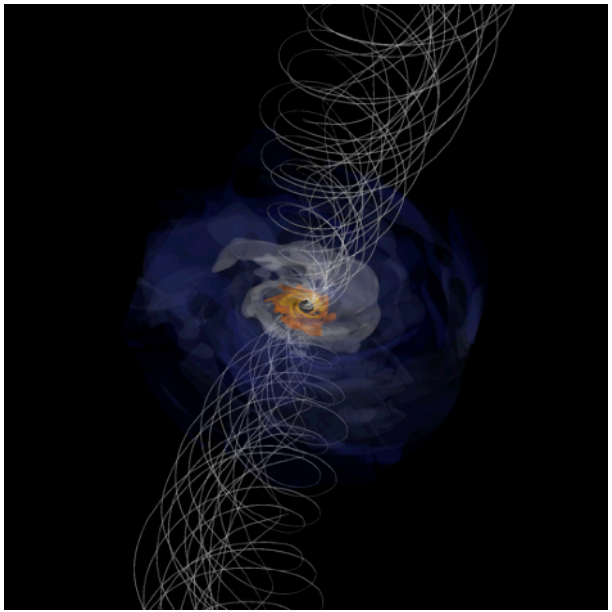


EHT Multi-wavelength group; ALMA, EVN; EAVN; VLBA; GMVA. Composition by J. C. Algaba

magnetic field lines advected with plasma

black holes drag spacetime as they **spin**

field twists around spin axis → Poynting flux



Wong+ 2025

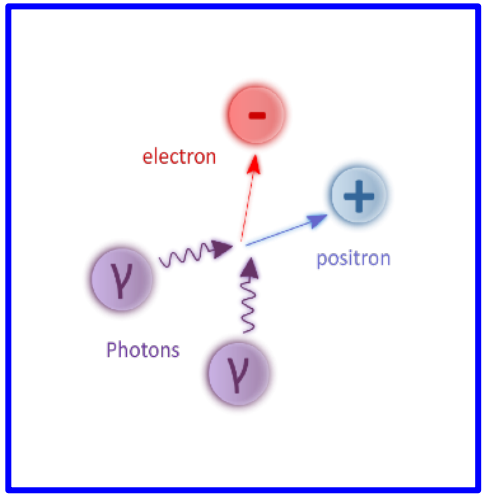
populating the jet

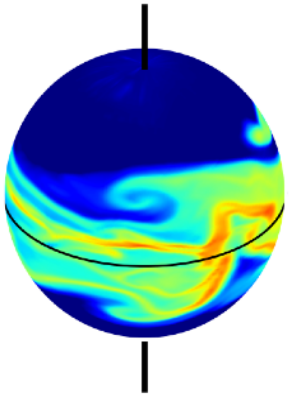
e.g., Beskin+, Hirotani & Okamoto, Ford+, Levinson & Cerutti, Chen+, Parfrey+

- electron–positron pair creation via gaps or drizzle

e.g., Mościbrodzka+, **Wong**+, Kimura & Toma, Yao+

- entrain plasma from the disk into the jet





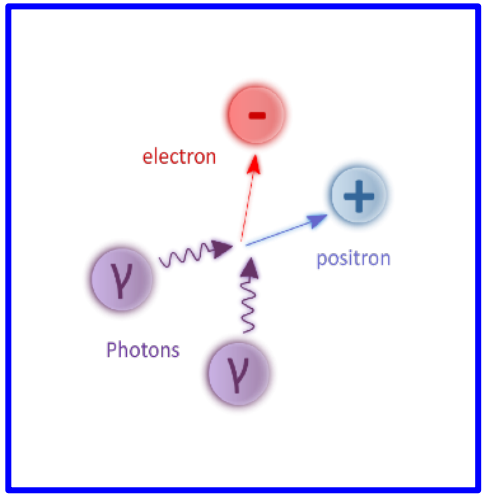
populating the jet

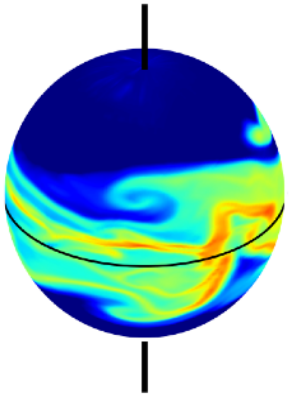
e.g., Beskin+, Hirotani & Okamoto, Ford+, Levinson & Cerutti, Chen+, Parfrey+

- **electron–positron pair creation via gaps or drizzle**

e.g., Mościbrodzka+, **Wong**+, Kimura & Toma, Yao+

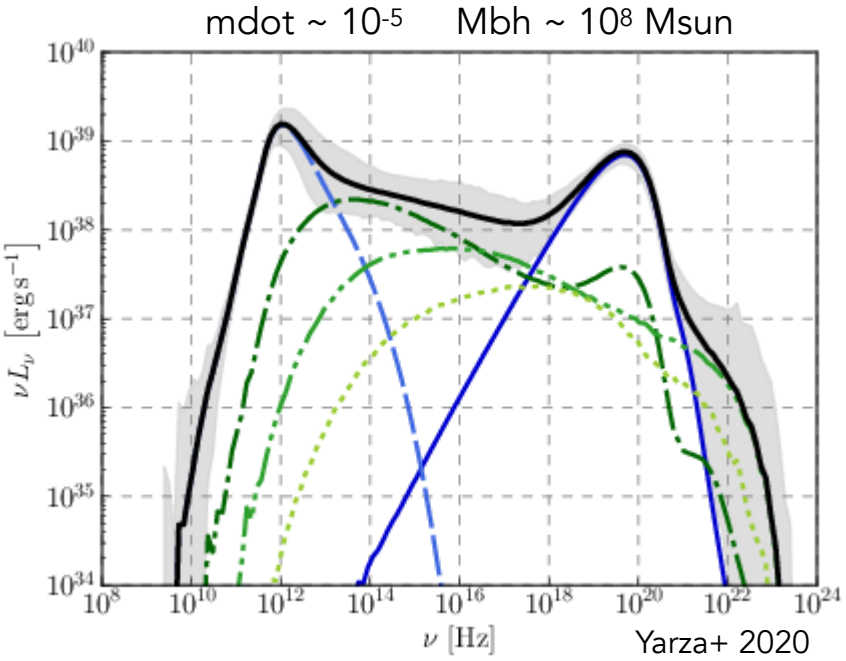
- entrain plasma from the disk into the jet





Breit–Wheeler process

- $e^- e^+$ pair created by **photon–photon** interaction
- photons → ≥ rest-mass energy of pair ~ 10^{21} Hz



pair cascade model

high frequency
+
low frequency

pair drizzle model

2 photons
near threshold

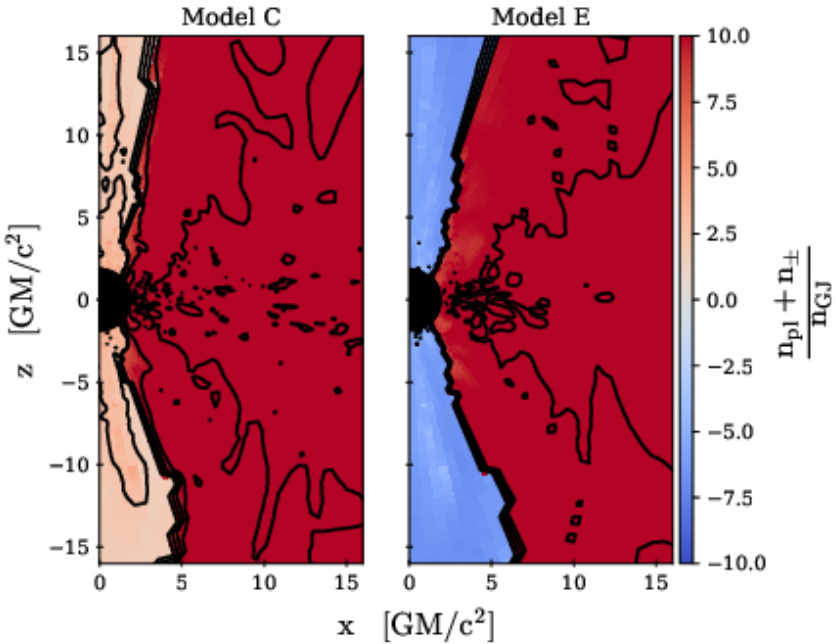
low charge regions → unscreened E fields
→ acceleration (Beskin+ 1992)

background radiation ← hot plasma
(Mościbrodzka+ 2011)

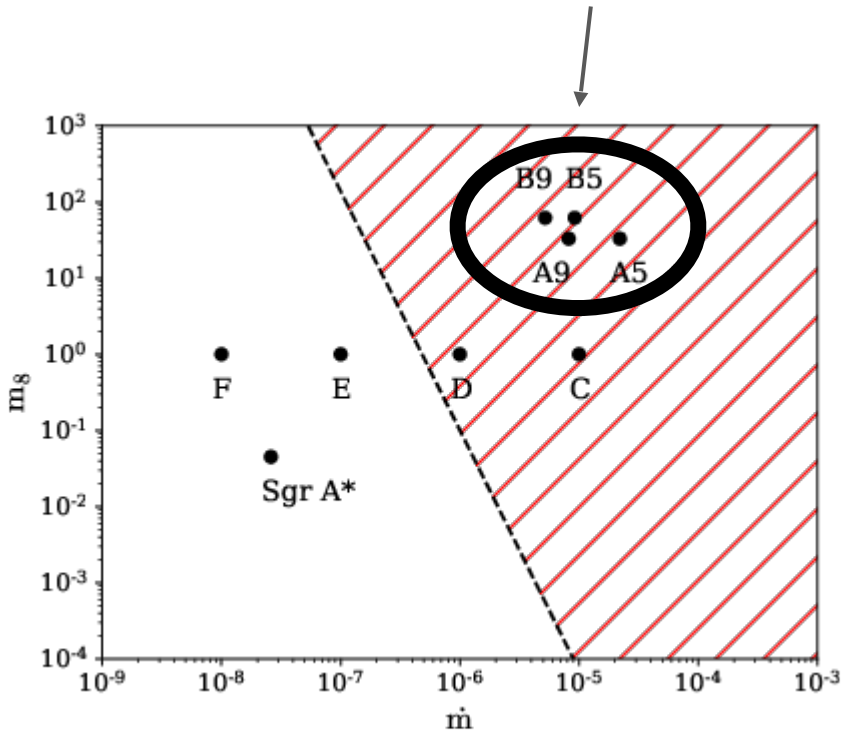
can pair cascades occur?

charge density is sufficient → no cascades?

insufficient charge to screen electric field → pair cascades?



M87 - like models



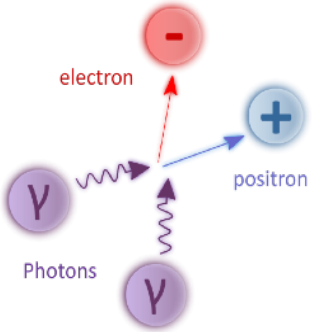
populating the jet

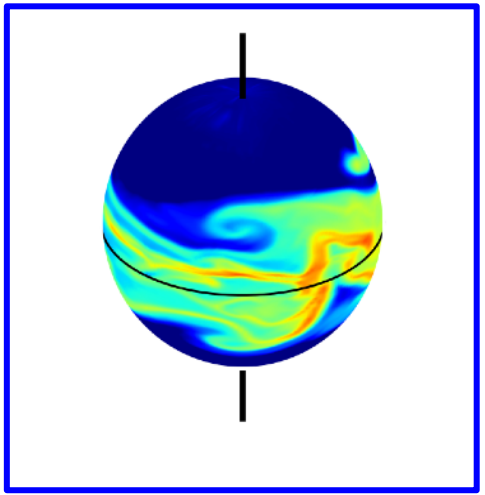
e.g., Beskin+, Hirotani & Okamoto, Ford+, Levinson & Cerutti, Chen+, Parfrey+

- electron–positron pair creation via gaps or drizzle

e.g., Mościbrodzka+, **Wong**+, Kimura & Toma, Yao+

- **entrain plasma from the disk into the jet**





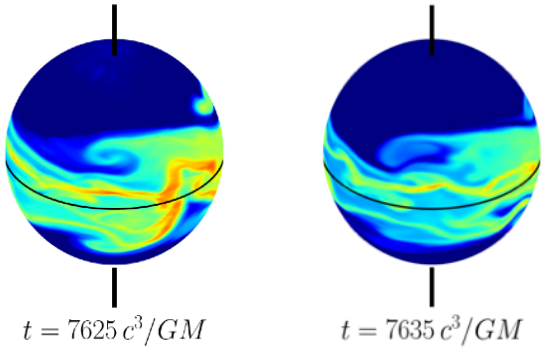
populating the jet & mass entrainment

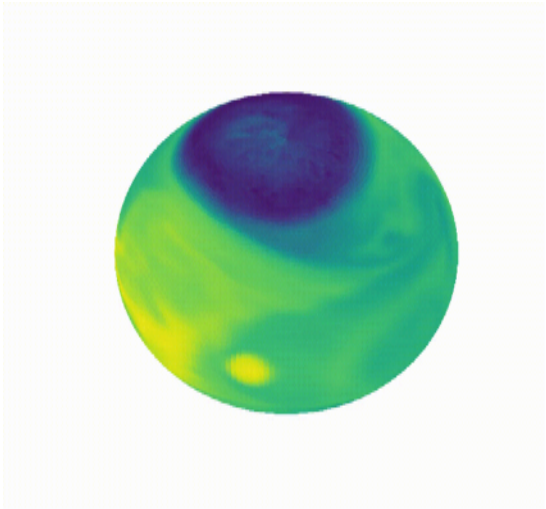
accretion flow anatomy

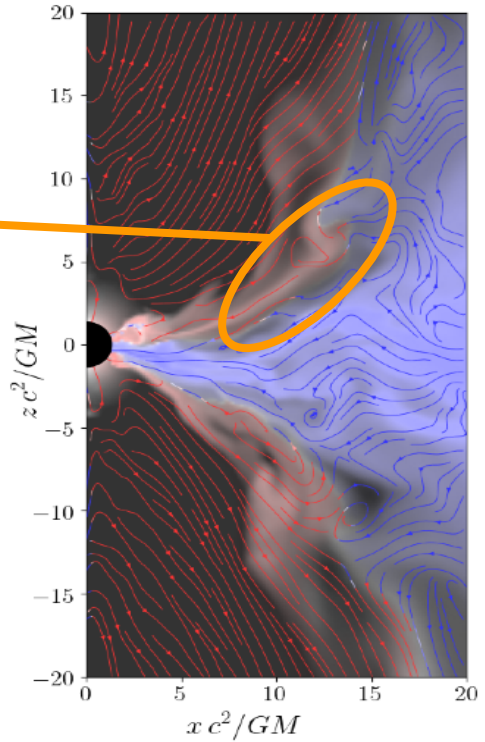
plasma flows **along magnetic field lines**

… but jet and disk are magnetically disconnected

jet–disk shear boundary layer **unstable**!







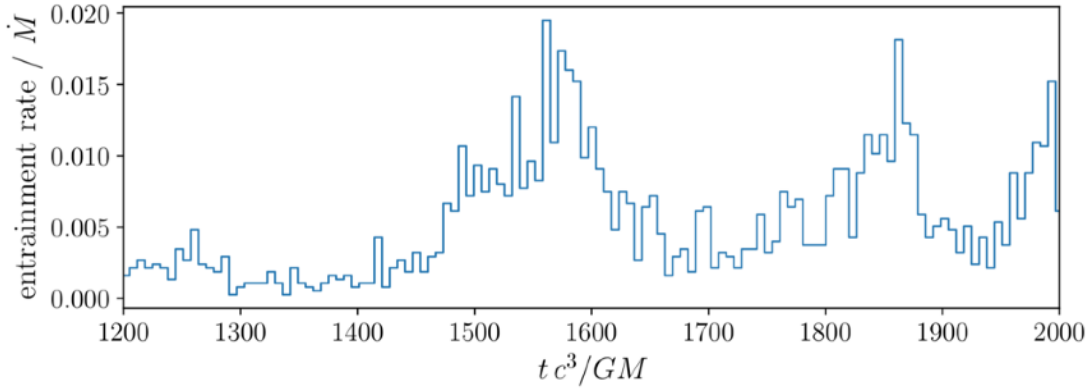
Wong+ 2021

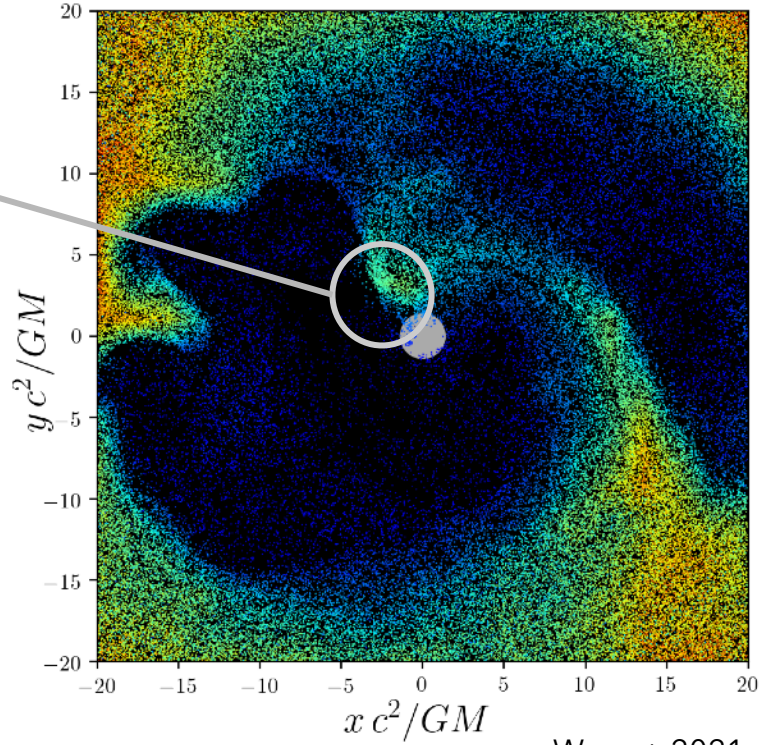
populating the jet & mass entrainment

measuring the entrainment rate

use tracer particles to track fluid motion onto the hole / are entrained into the jet

tracers are entrained at **Kelvin–Helmholtz rolls** and **magnetic torque events** (flux tube interactions) near the horizon





Wong+ 2021

What Lagrangian question does your project need to answer?

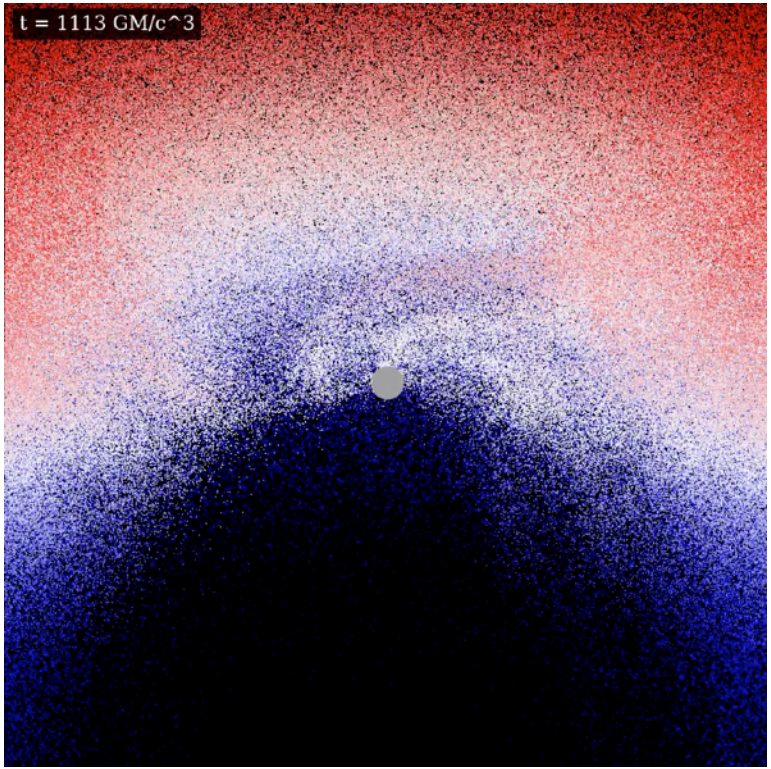
Particles can **add Lagrangian state** to grid simulations.

> Passive particles record histories.
> Active particles exchange mass, momentum, energy, … with the mesh.

Current AthenaK particle implementations are **mostly passive**:
gather -> push -> communicate -> output

Using particles well **requires practical checks**:
what state is carried?
what pusher is used?
what happens at boundaries / AMR / restart?
what statistic should converge?

Tracer science:
measure advection, mixing, residence time, and origin



Applications:
> SANE/MAD disks transport mass differently, **the initial condition matters**
> jet loading can come from **disk-jet boundary entrainment**

Thank you!