Astrophysical Multi-Messenger Modeling

AM^3: An Open-Source Tool for Time-Dependent Lepto-Hadronic Modeling of Astrophysical Sources

Marc Klinger-Plaisier*, 14.10.2024,
GRAPPA journal club

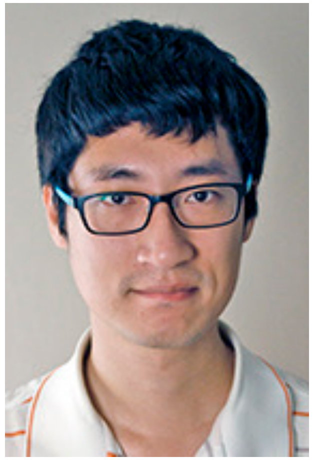
arXiv:2312.13371

HELMHOLTZ WEIZMANN RESEARCH SCHOOL MULTIMESSENGER ASTRONOMY
HELMHOLTZ
DESY.

*m.klinger@uva.nl
https://maklinger.github.io/

Universiteit van Amsterdam
Anton Pannekoek Instituut

The AM³ team

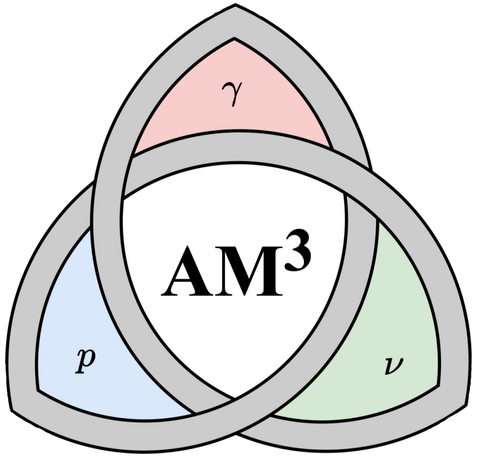


Gao

Klinger

Rudolph

Rodrigues



Yuan

Fichet De Clairfontaine

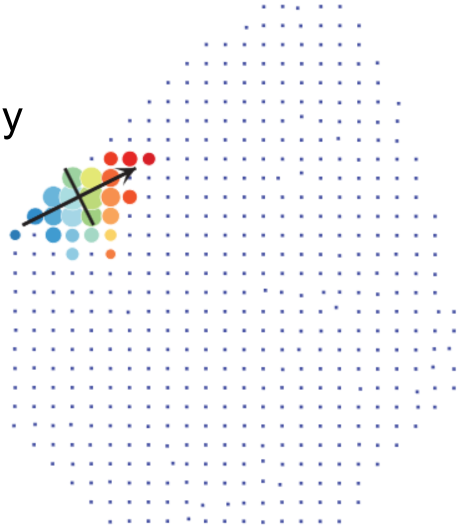
Fedynitch

Winter

Pohl

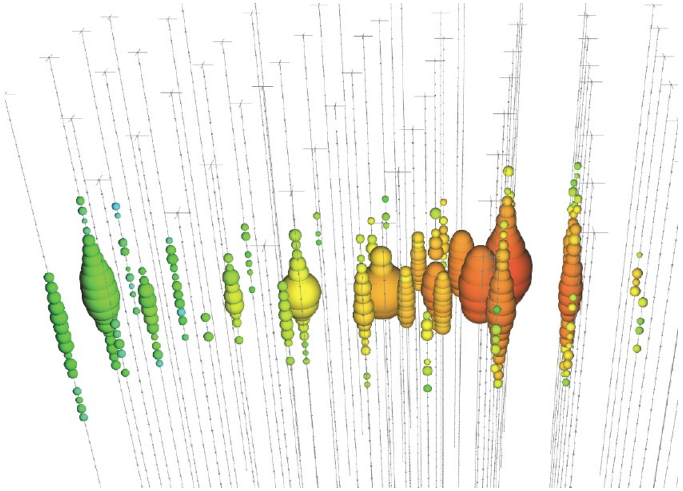
Typical multi-messenger astrophysics challenge

cosmic ray

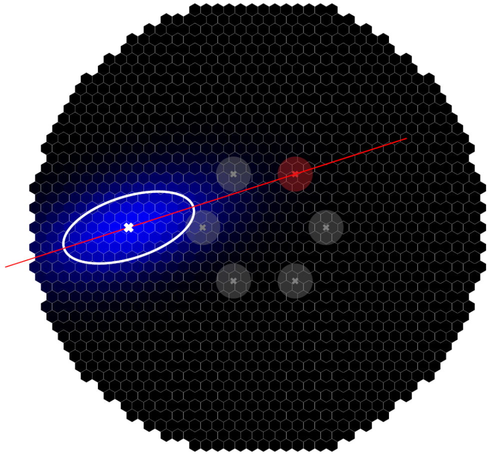


TA collaboration 2023

neutrino



IceCube Masterclass

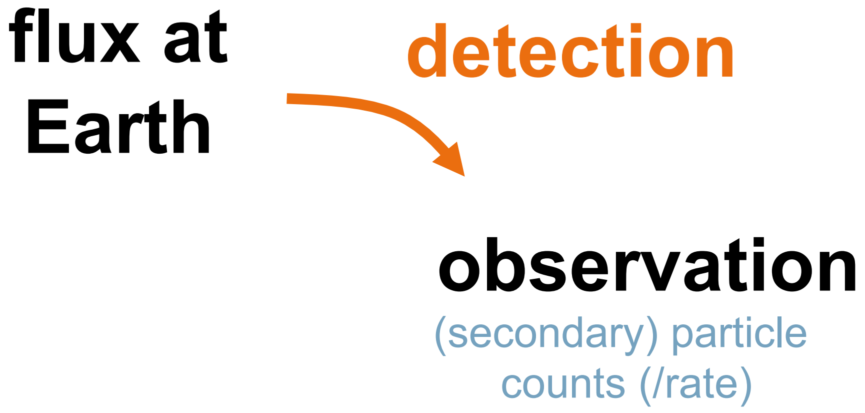


γ-ray

observation

(secondary) particle counts (/rate)

Typical multi-messenger astrophysics challenge



flux at Earth

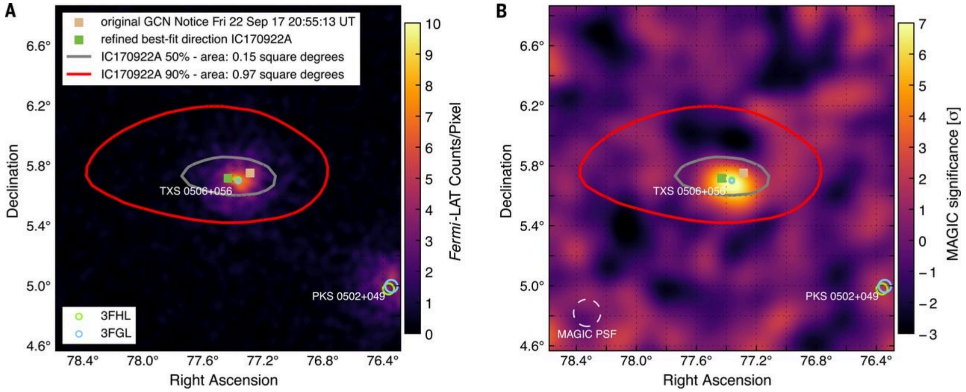
detection

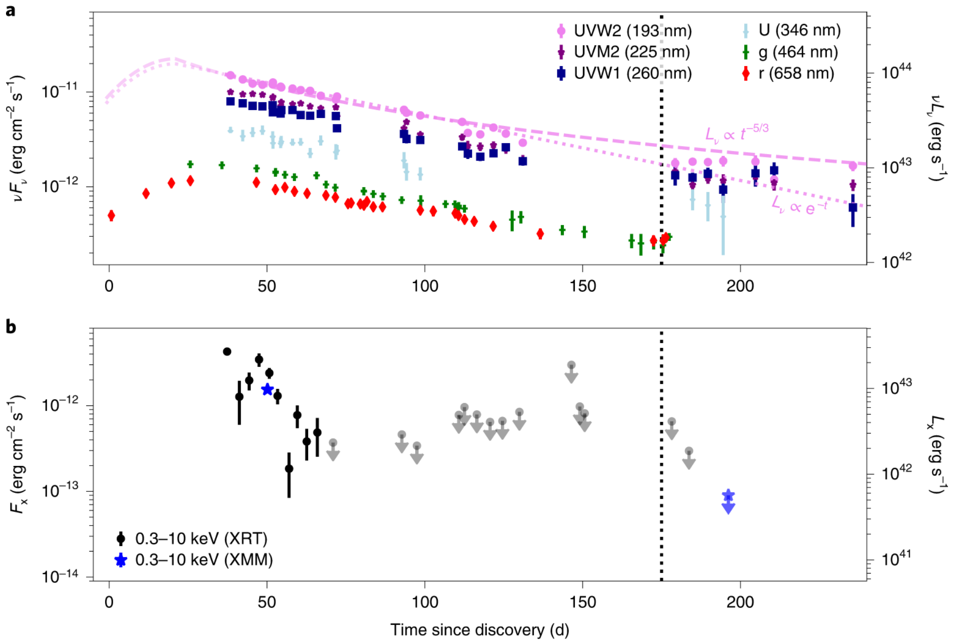
observation

(secondary) particle counts (/rate)

Typical multi-messenger astrophysics challenge

IceCube et al., Science, 361,146 (2018)

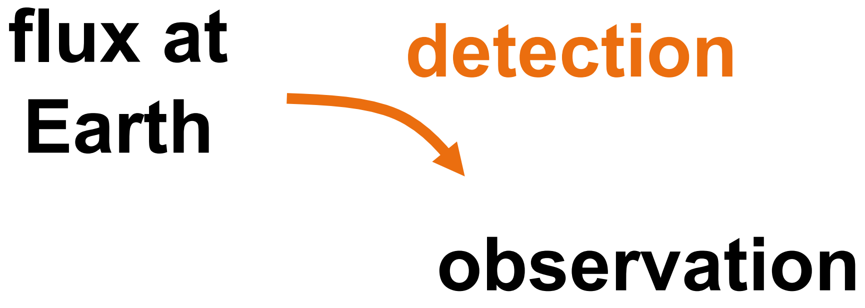




Stein et al., Nature Astronomy, 5, 510-518 (2021)

spatial/temporal coincidences

e.g. blazars, tidal disruption events (TDE)



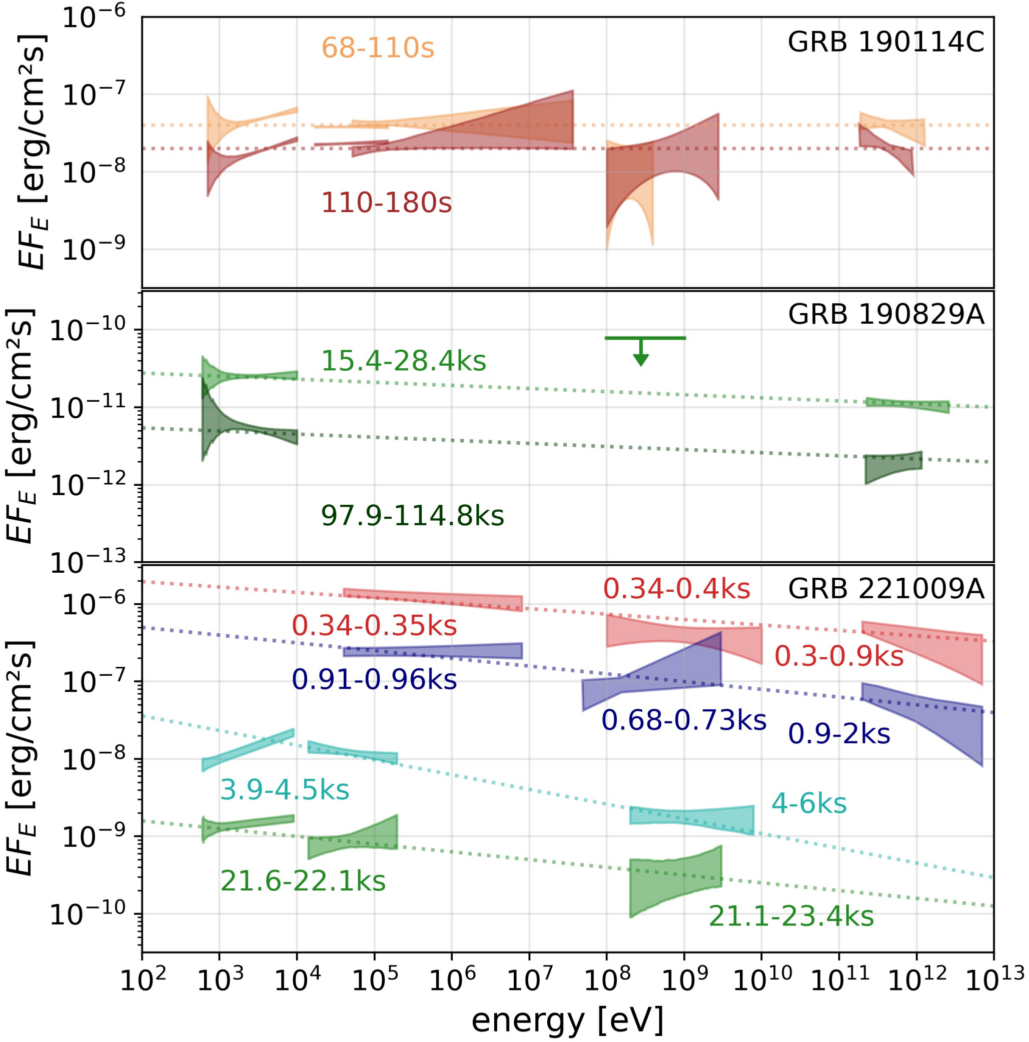
flux at Earth

detection

observation

Typical multi-messenger astrophysics challenge

MK++ arXiv:2403.13902



spatial/temporal coincidences

e.g. blazars, tidal disruption events (TDEs)

multi-wavelength spectra

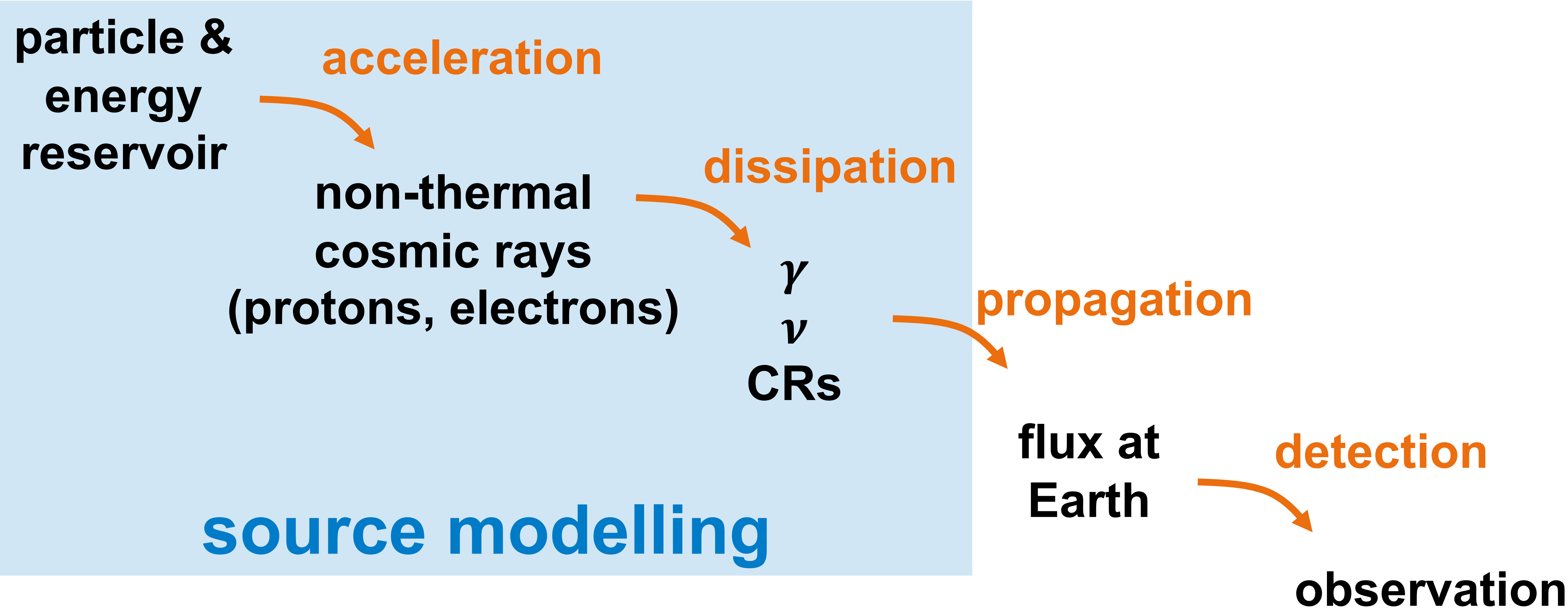
e.g. blazars, gamma-ray bursts (GRBs),…

flux at Earth

detection

observation

Typical multi-messenger astrophysics challenge



particle & energy reservoir

acceleration

non-thermal cosmic rays (protons, electrons)

dissipation

γ

ν

CRs

propagation

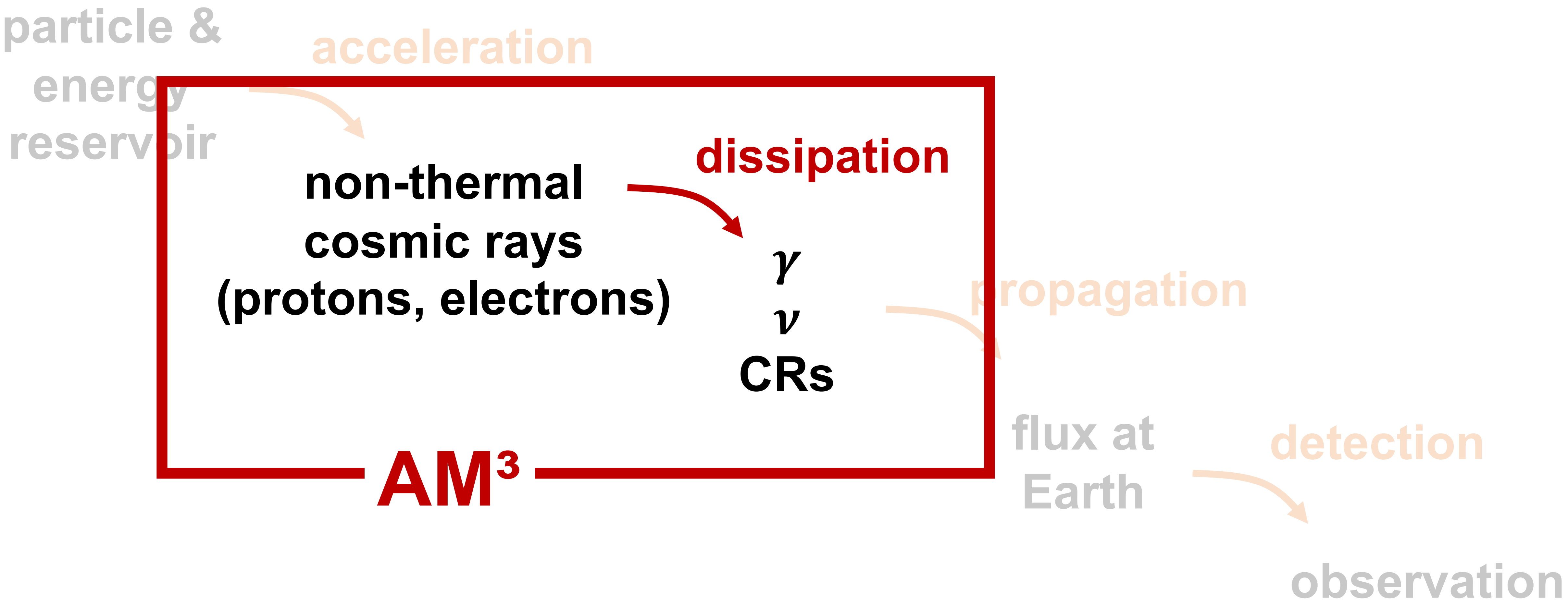
flux at Earth

detection

observation

source modelling

Typical multi-messenger astrophysics challenge



particle & energy reservoir

acceleration

non-thermal cosmic rays (protons, electrons)

dissipation

γ

ν

CRs

AM³

propagation

flux at Earth

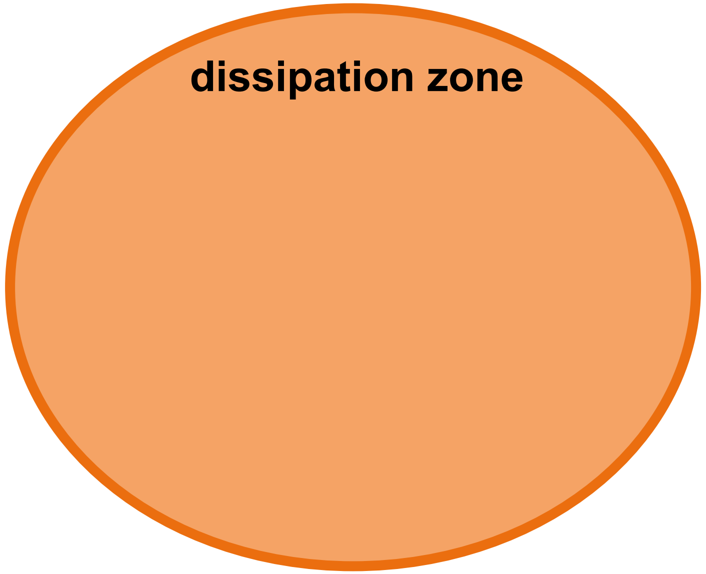
detection

observation

Astrophysical Multi-Messenger Modeling

The task

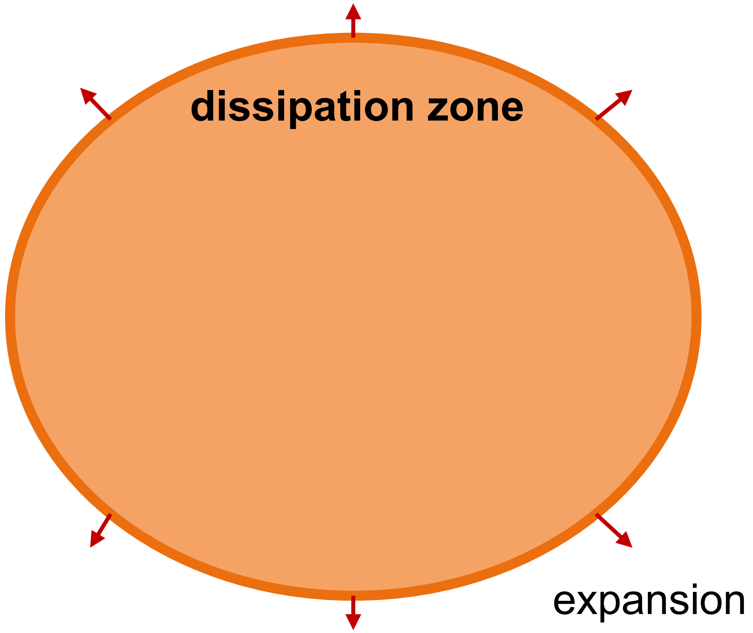
→ in the comoving frame
→ homogeneous/isotropic



dissipation zone

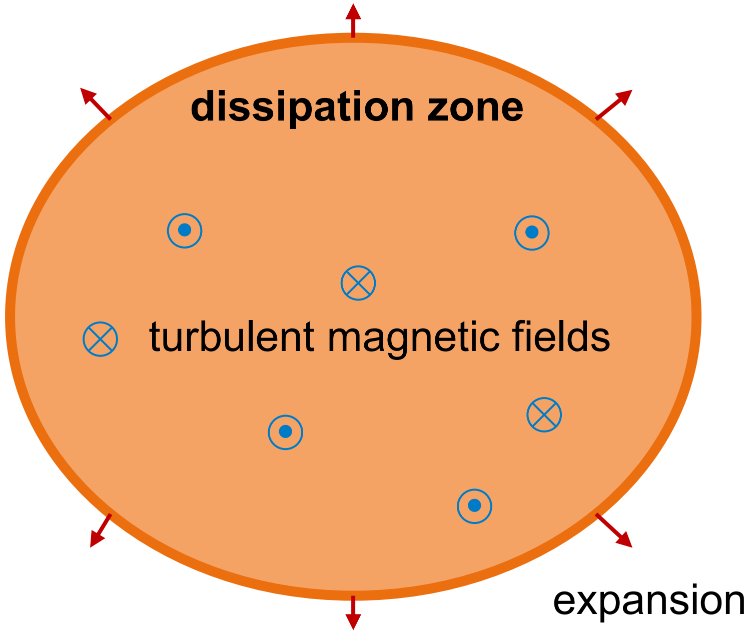
The task

→ in the comoving frame
→ homogeneous/isotropic



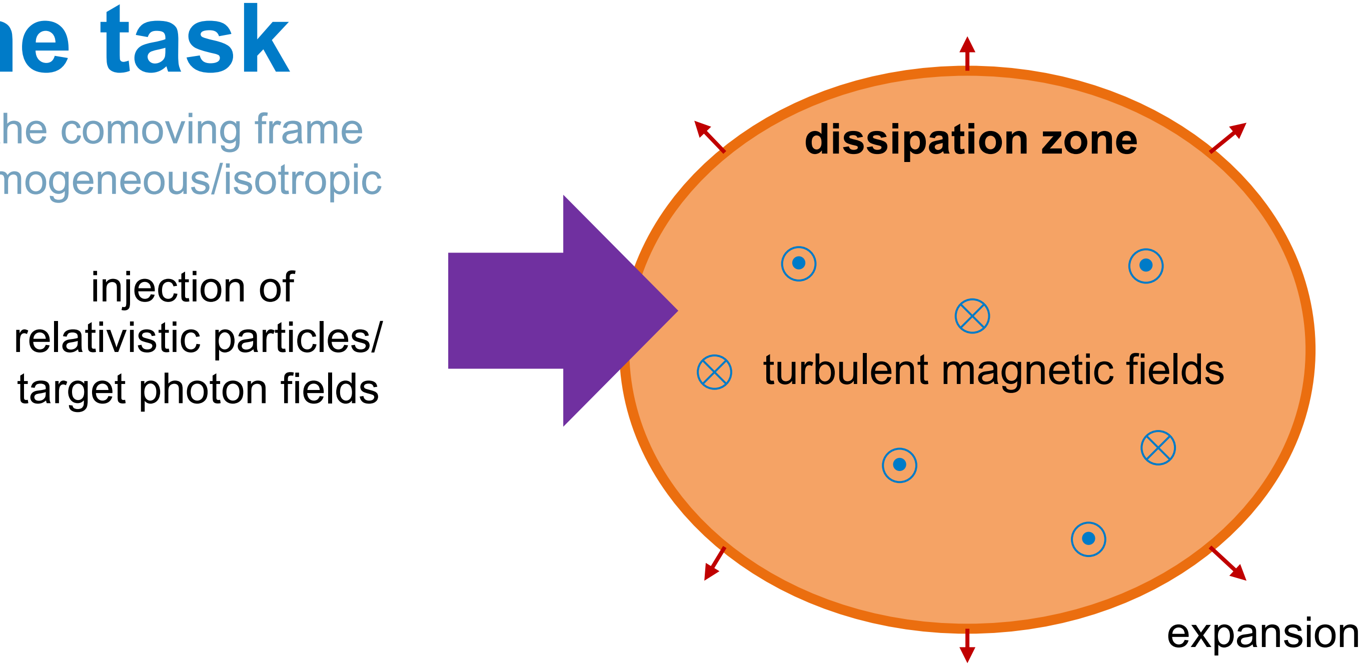
The task

→ in the comoving frame
→ homogeneous/isotropic



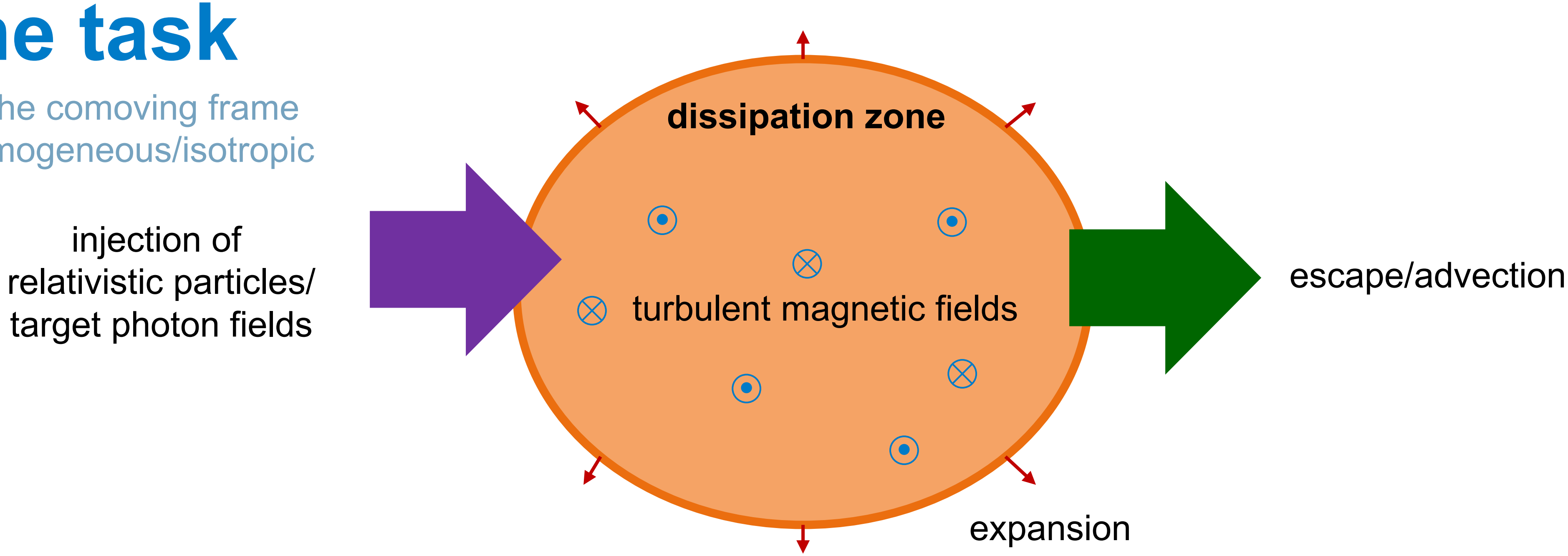
The task

→ in the comoving frame
→ homogeneous/isotropic



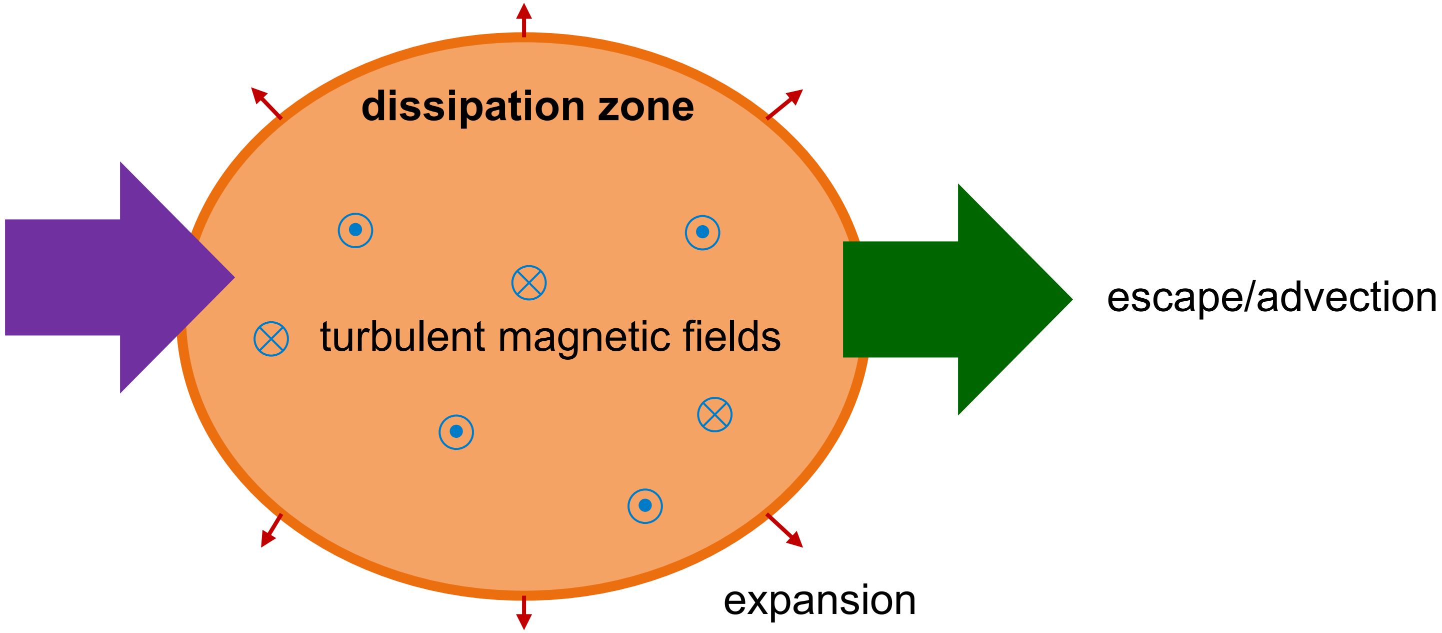
The task

→ in the comoving frame
→ homogeneous/isotropic



The task

→ in the comoving frame
→ homogeneous/isotropic


injection of relativistic particles/ target photon fields
dissipation zone
turbulent magnetic fields
escape/advection
expansion

→ solve transport eqs.

$$\partial_t n_i = Q + \partial_E (\dot{E} n_i) - \alpha n_i$$

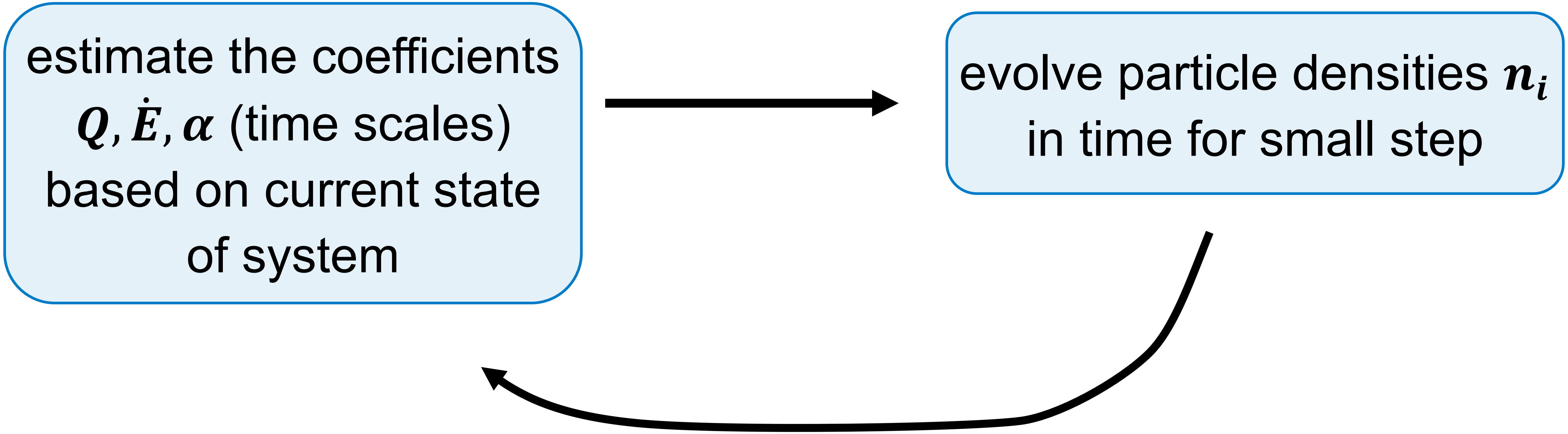
for species $i \in [p, n, e, \pi, \mu, \nu, \gamma]$

particle number density

$$n_i(E,t) = \frac{\partial^2 N_i}{\partial E\ \partial V}$$

depend in general on E, t, n_j

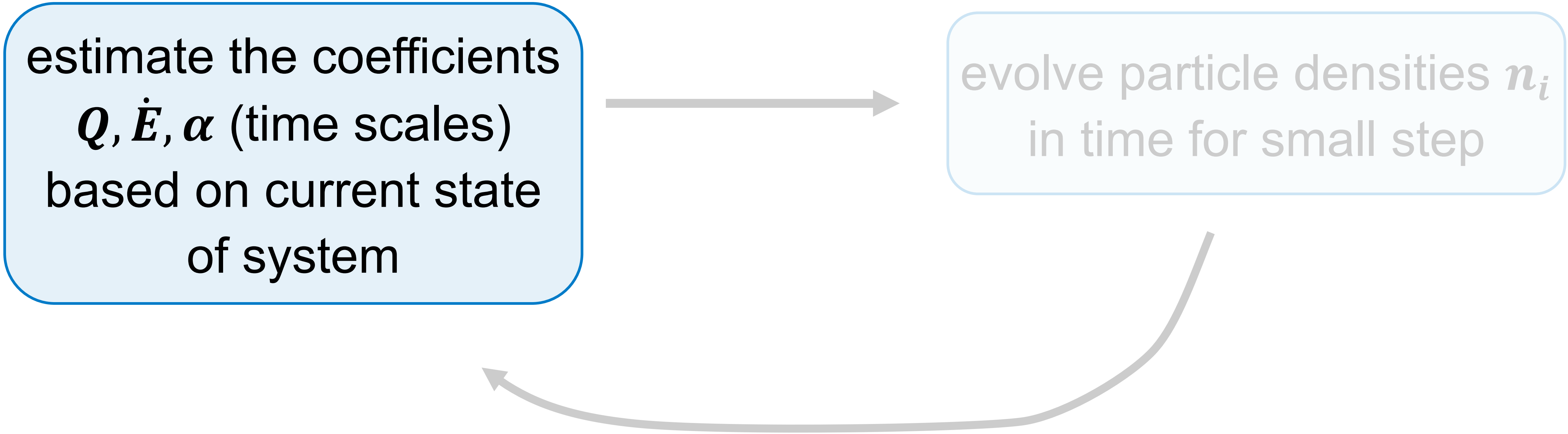
The workflow



estimate the coefficients $\boldsymbol{Q}, \boldsymbol{\dot{E}}, \boldsymbol{\alpha}$ (time scales) based on current state of system

evolve particle densities $\boldsymbol{n_i}$ in time for small step

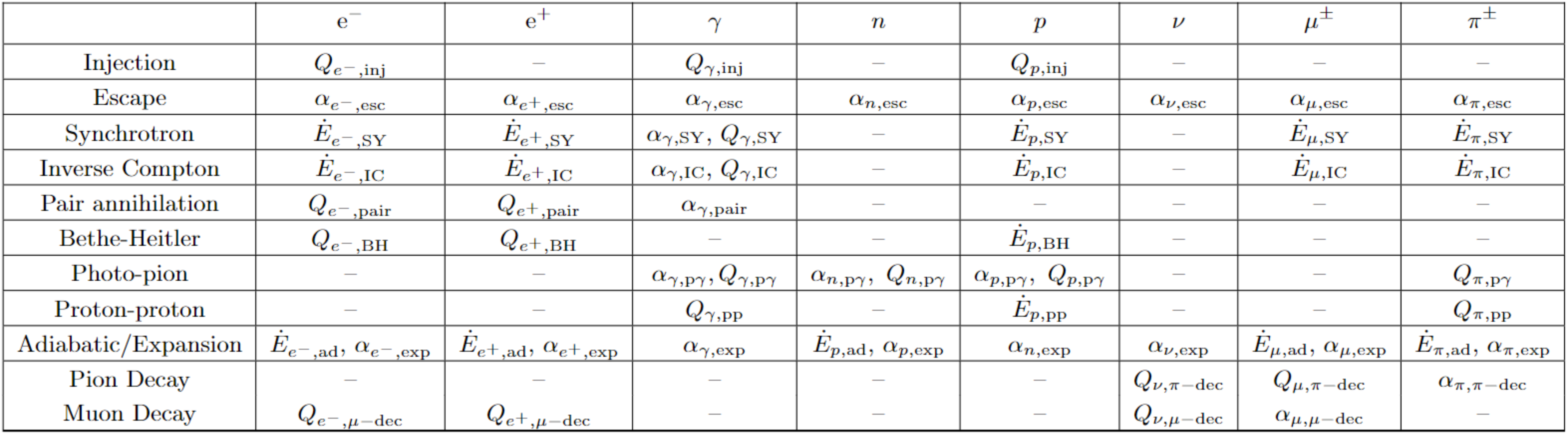
The workflow



estimate the coefficients $\boldsymbol{Q}, \dot{\boldsymbol{E}}, \boldsymbol{\alpha}$ (time scales) based on current state of system

evolve particle densities $\boldsymbol{n_i}$ in time for small step

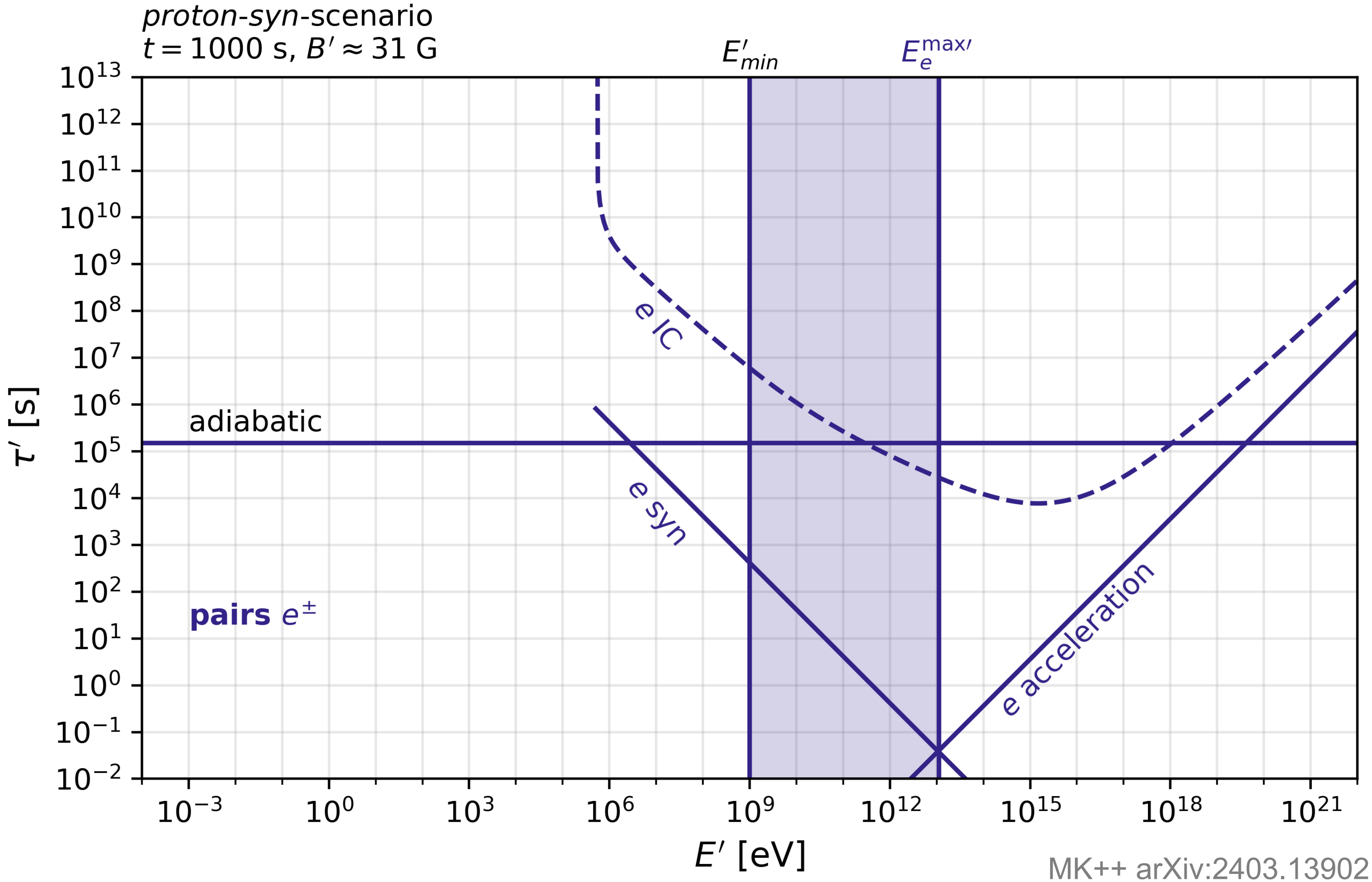
Estimate the coefficients



	e$^-$	e$^+$	γ	n	p	ν	$\mu^\pm$	$\pi^\pm$
Injection	$Q_{e^-,\mathrm{inj}}$	–	$Q_{\gamma,\mathrm{inj}}$	–	$Q_{p,\mathrm{inj}}$	–	–	–
Escape	$\alpha_{e^-,\mathrm{esc}}$	$\alpha_{e^+,\mathrm{esc}}$	$\alpha_{\gamma,\mathrm{esc}}$	$\alpha_{n,\mathrm{esc}}$	$\alpha_{p,\mathrm{esc}}$	$\alpha_{\nu,\mathrm{esc}}$	$\alpha_{\mu,\mathrm{esc}}$	$\alpha_{\pi,\mathrm{esc}}$
Synchrotron	$\dot{E}_{e^-,\mathrm{SY}}$	$\dot{E}_{e^+,\mathrm{SY}}$	$\alpha_{\gamma,\mathrm{SY}}$, $Q_{\gamma,\mathrm{SY}}$	–	$\dot{E}_{p,\mathrm{SY}}$	–	$\dot{E}_{\mu,\mathrm{SY}}$	$\dot{E}_{\pi,\mathrm{SY}}$
Inverse Compton	$\dot{E}_{e^-,\mathrm{IC}}$	$\dot{E}_{e^+,\mathrm{IC}}$	$\alpha_{\gamma,\mathrm{IC}}$, $Q_{\gamma,\mathrm{IC}}$	–	$\dot{E}_{p,\mathrm{IC}}$	–	$\dot{E}_{\mu,\mathrm{IC}}$	$\dot{E}_{\pi,\mathrm{IC}}$
Pair annihilation	$Q_{e^-,\mathrm{pair}}$	$Q_{e^+,\mathrm{pair}}$	$\alpha_{\gamma,\mathrm{pair}}$	–	–	–	–	–
Bethe-Heitler	$Q_{e^-,\mathrm{BH}}$	$Q_{e^+,\mathrm{BH}}$	–	–	$\dot{E}_{p,\mathrm{BH}}$	–	–	–
Photo-pion	–	–	$\alpha_{\gamma,\mathrm{p}\gamma}$, $Q_{\gamma,\mathrm{p}\gamma}$	$\alpha_{n,\mathrm{p}\gamma}$, $Q_{n,\mathrm{p}\gamma}$	$\alpha_{p,\mathrm{p}\gamma}$, $Q_{p,\mathrm{p}\gamma}$	–	–	$Q_{\pi,\mathrm{p}\gamma}$
Proton-proton	–	–	$Q_{\gamma,\mathrm{pp}}$	–	$\dot{E}_{p,\mathrm{pp}}$	–	–	$Q_{\pi,\mathrm{pp}}$
Adiabatic/Expansion	$\dot{E}_{e^-,\mathrm{ad}}$, $\alpha_{e^-,\mathrm{exp}}$	$\dot{E}_{e^+,\mathrm{ad}}$, $\alpha_{e^+,\mathrm{exp}}$	$\alpha_{\gamma,\mathrm{exp}}$	$\dot{E}_{p,\mathrm{ad}}$, $\alpha_{p,\mathrm{exp}}$	$\alpha_{n,\mathrm{exp}}$	$\alpha_{\nu,\mathrm{exp}}$	$\dot{E}_{\mu,\mathrm{ad}}$, $\alpha_{\mu,\mathrm{exp}}$	$\dot{E}_{\pi,\mathrm{ad}}$, $\alpha_{\pi,\mathrm{exp}}$
Pion Decay	–	–	–	–	–	$Q_{\nu,\pi-\mathrm{dec}}$	$Q_{\mu,\pi-\mathrm{dec}}$	$\alpha_{\pi,\pi-\mathrm{dec}}$
Muon Decay	$Q_{e^-,\mu-\mathrm{dec}}$	$Q_{e^+,\mu-\mathrm{dec}}$	–	–	–	$Q_{\nu,\mu-\mathrm{dec}}$	$\alpha_{\mu,\mu-\mathrm{dec}}$	–

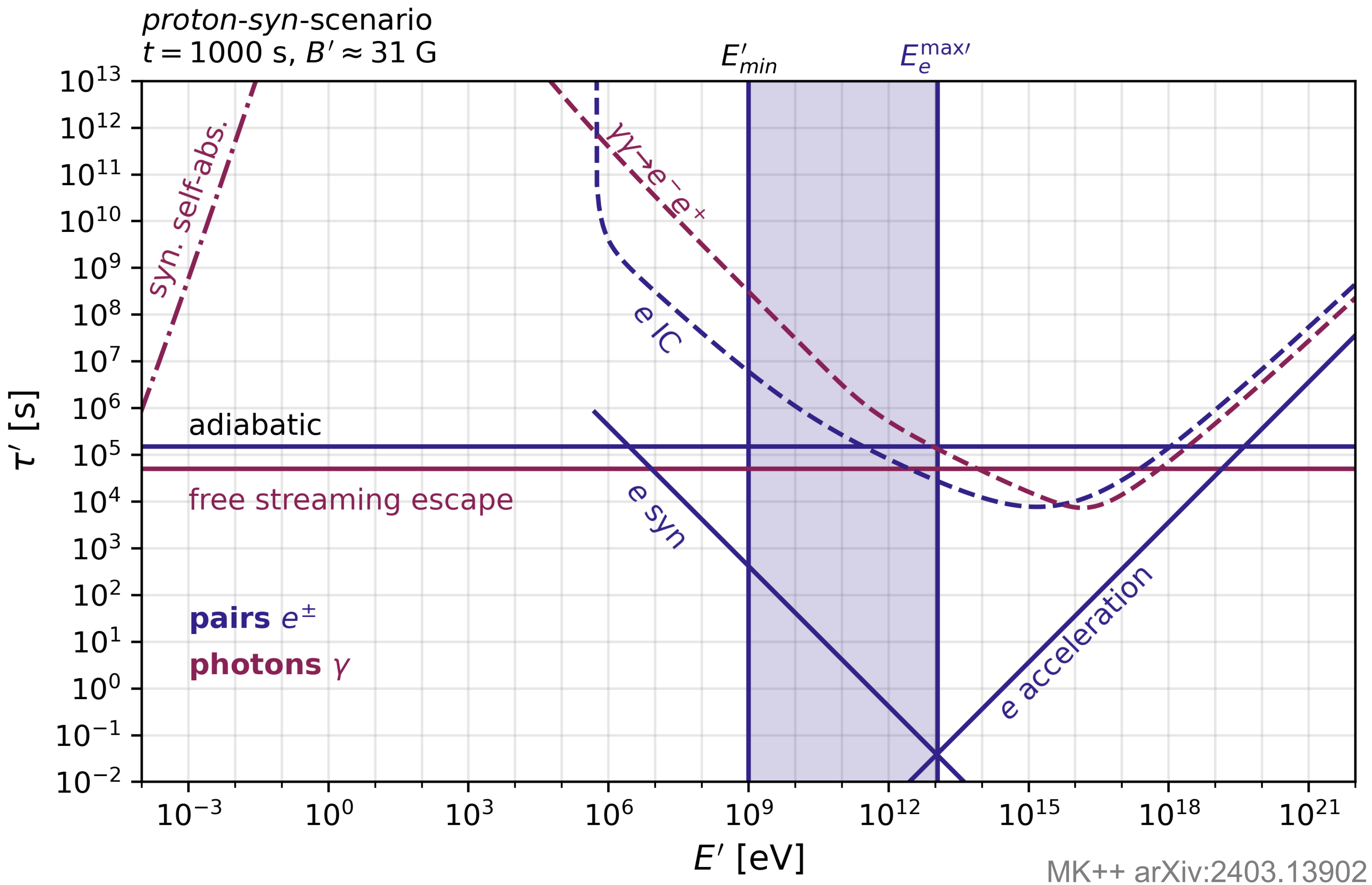
→ see appendix of arxiv:2312.13371 for details

Estimate the coefficients: GRB example



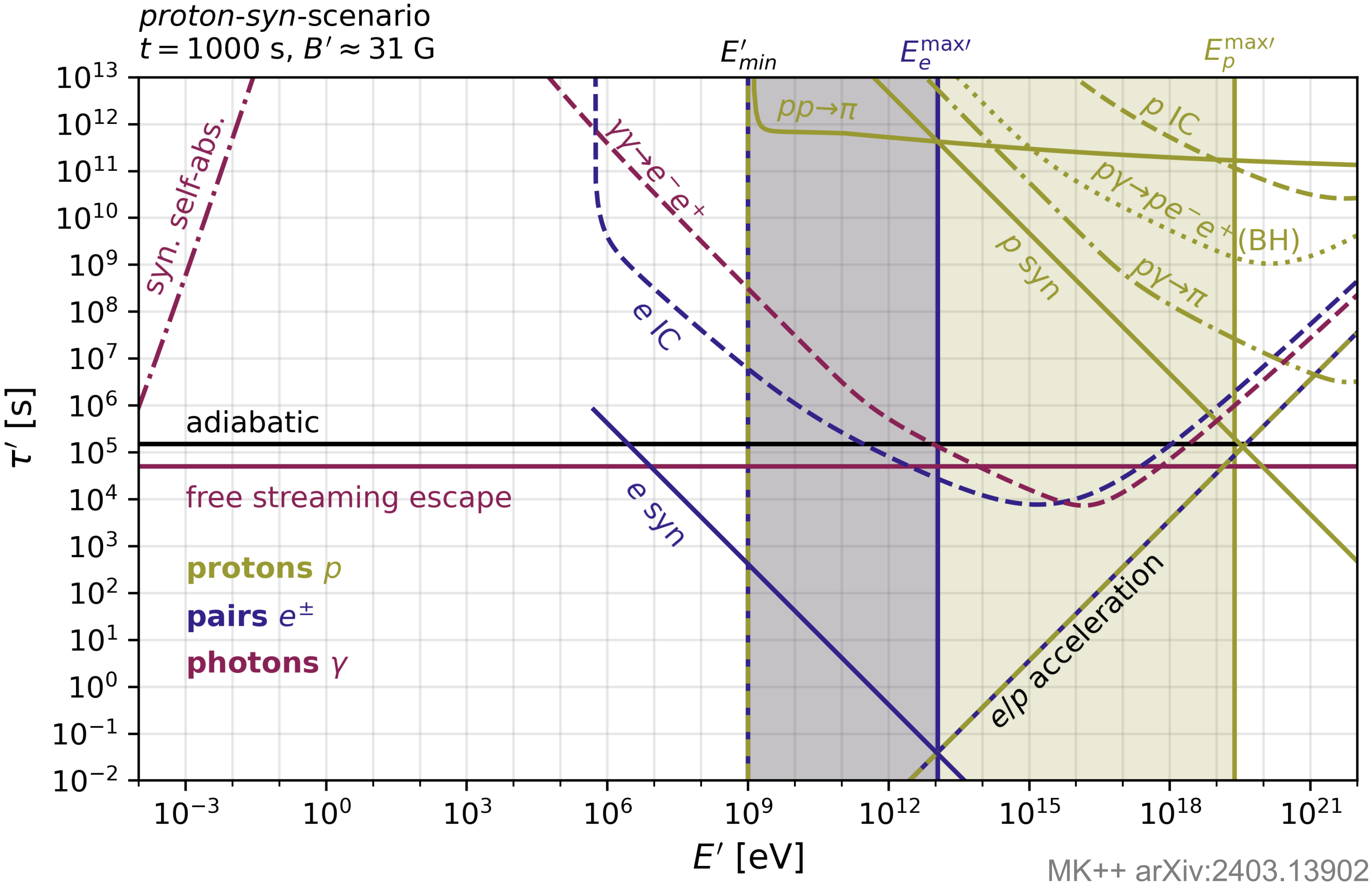
MK++ arXiv:2403.13902

Estimate the coefficients: GRB example



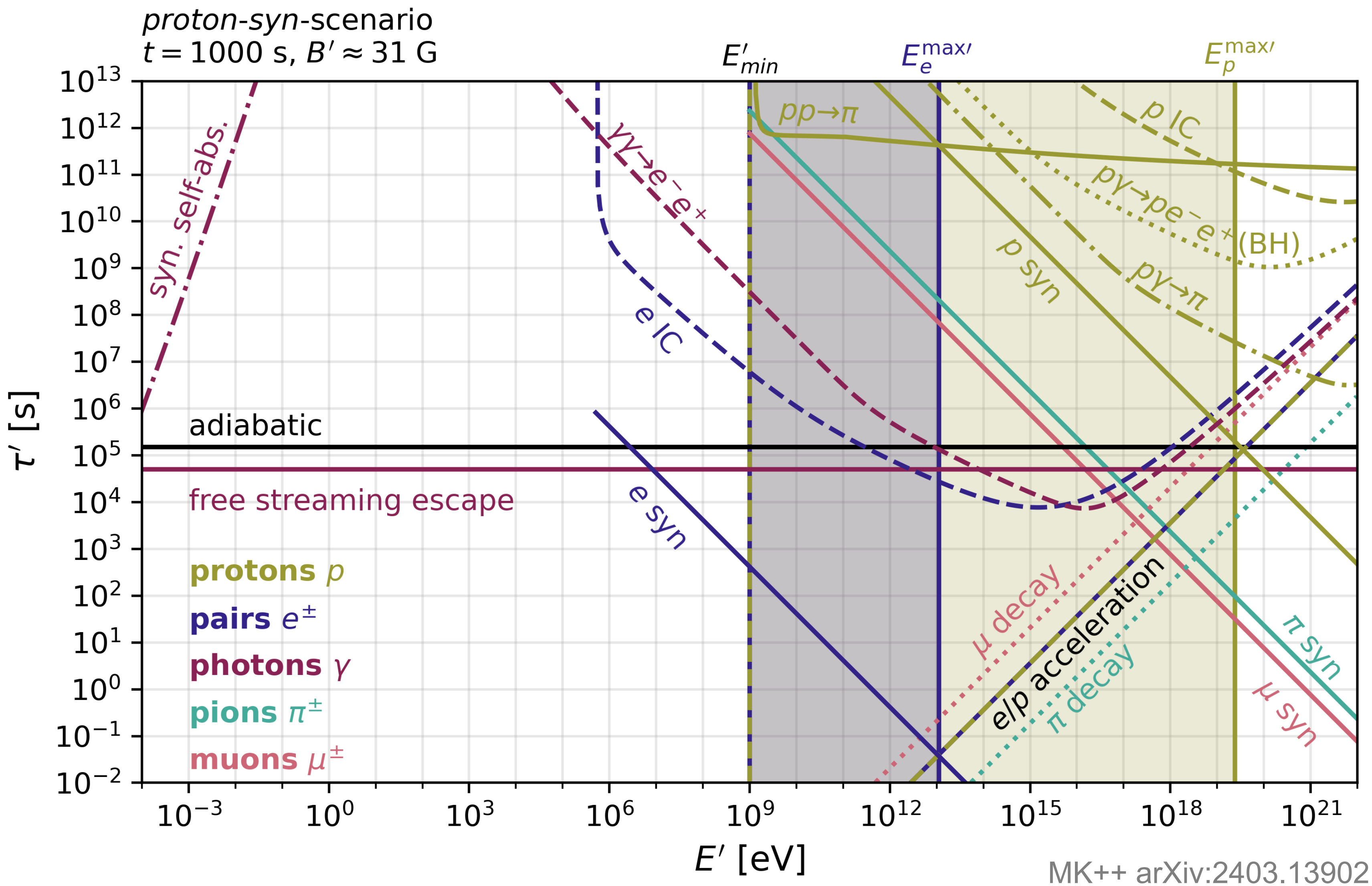
MK++ arXiv:2403.13902

Estimate the coefficients: GRB example



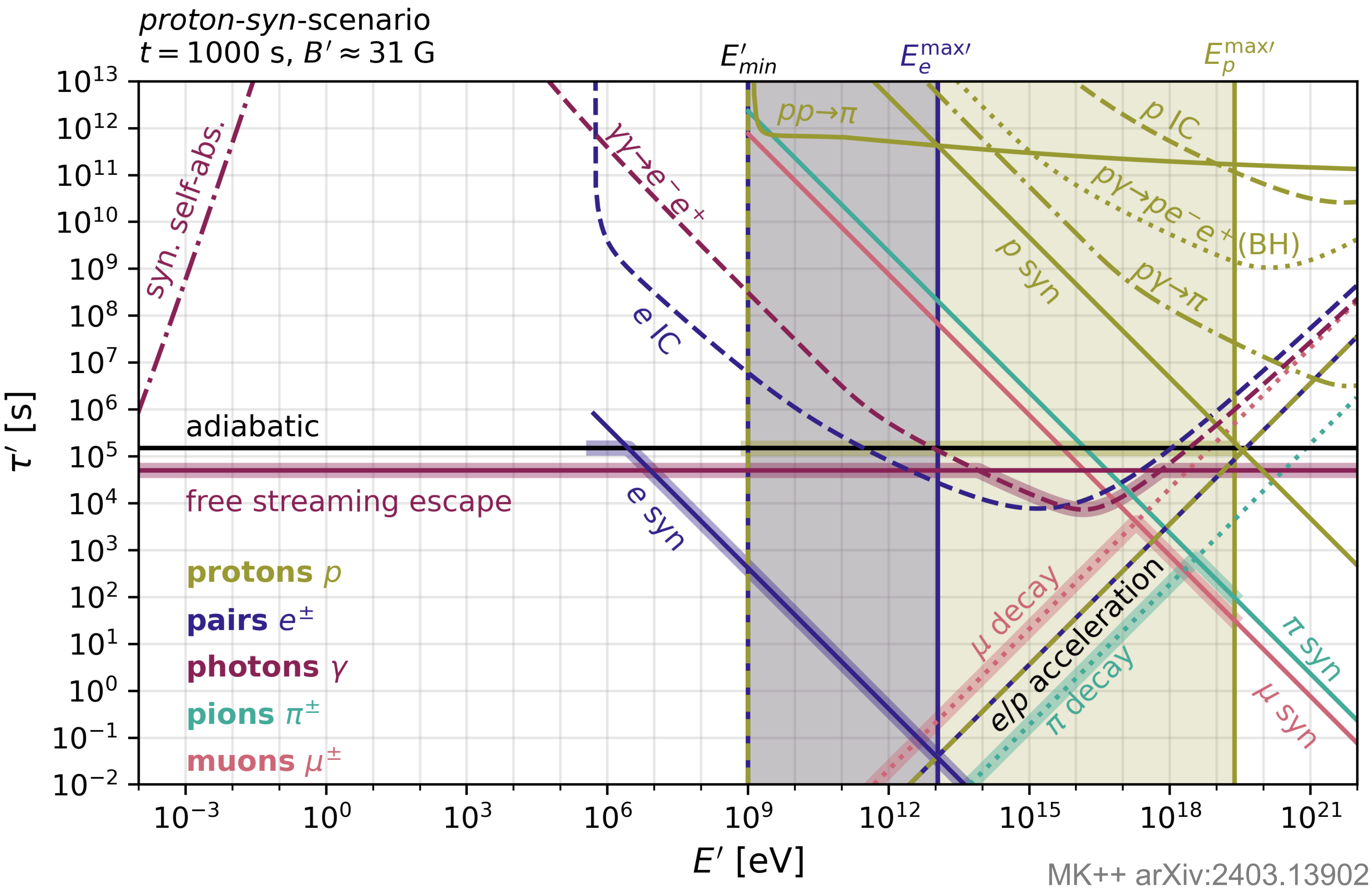
MK++ arXiv:2403.13902

Estimate the coefficients: GRB example



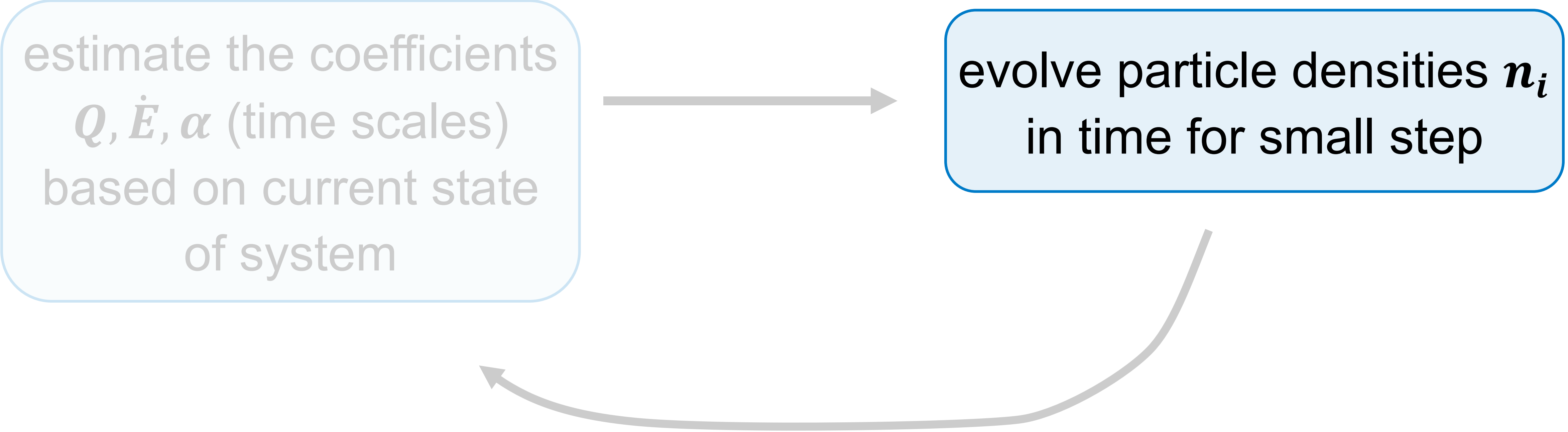
MK++ arXiv:2403.13902

Estimate the coefficients: GRB example



MK++ arXiv:2403.13902

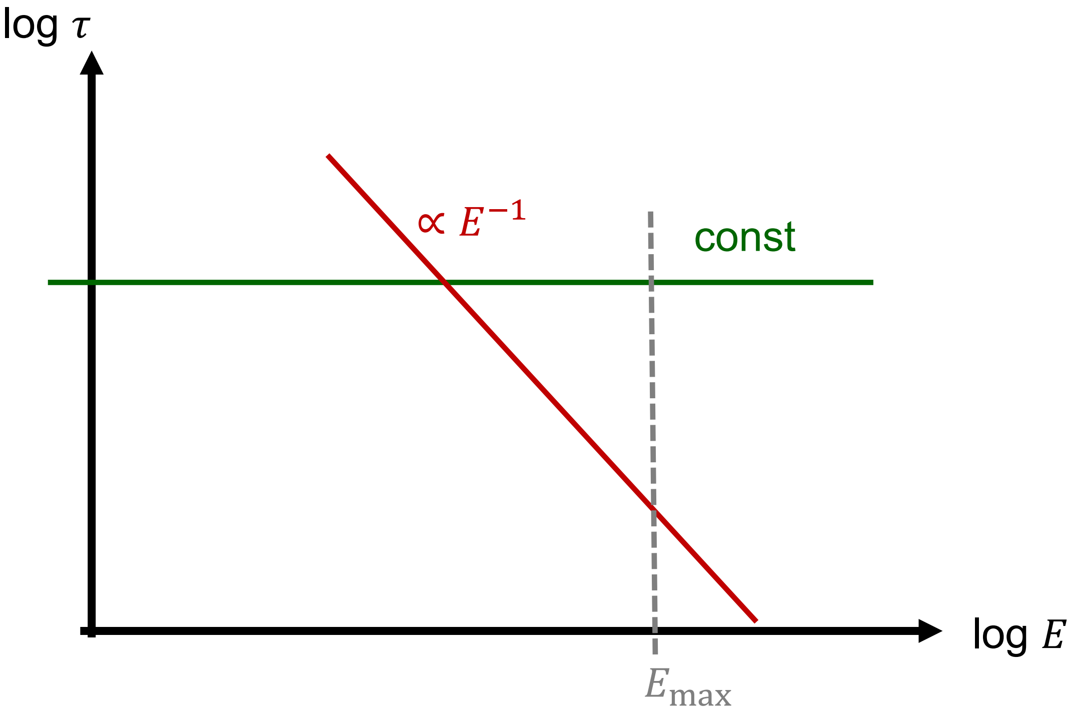
The workflow



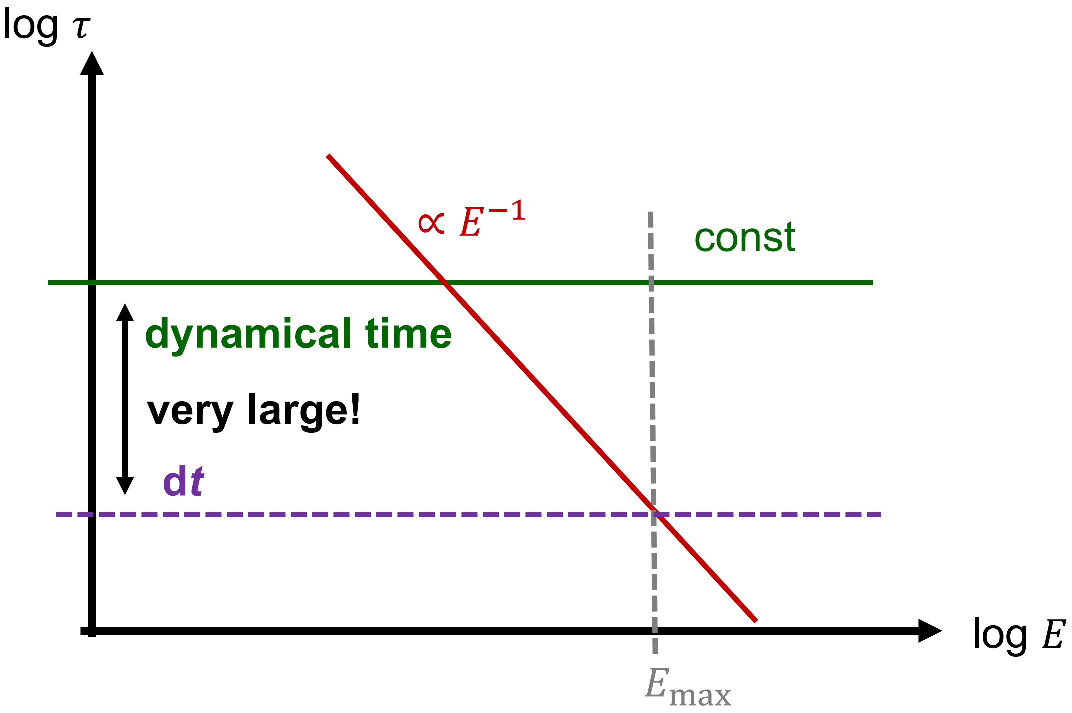
estimate the coefficients $\boldsymbol{Q}, \dot{\boldsymbol{E}}, \boldsymbol{\alpha}$ (time scales) based on current state of system

evolve particle densities $\boldsymbol{n_i}$ in time for small step

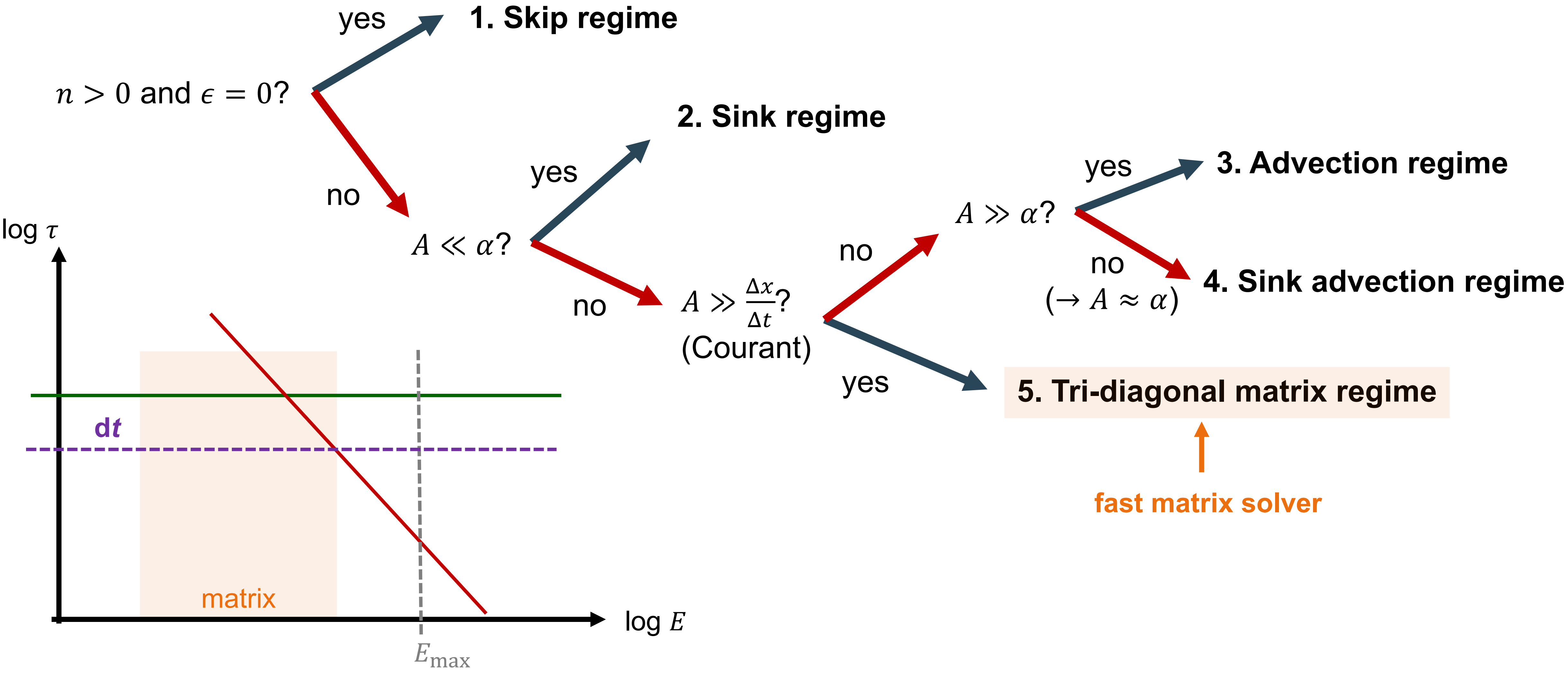
Evolve the particle densities



Evolve the particle densities



Evolve the particle densities



$n > 0$ and $\epsilon = 0$?

yes → **1. Skip regime**

no → $A \ll \alpha$?

yes → **2. Sink regime**

no → $A \gg \frac{\Delta x}{\Delta t}$? (Courant)

no → $A \gg \alpha$?

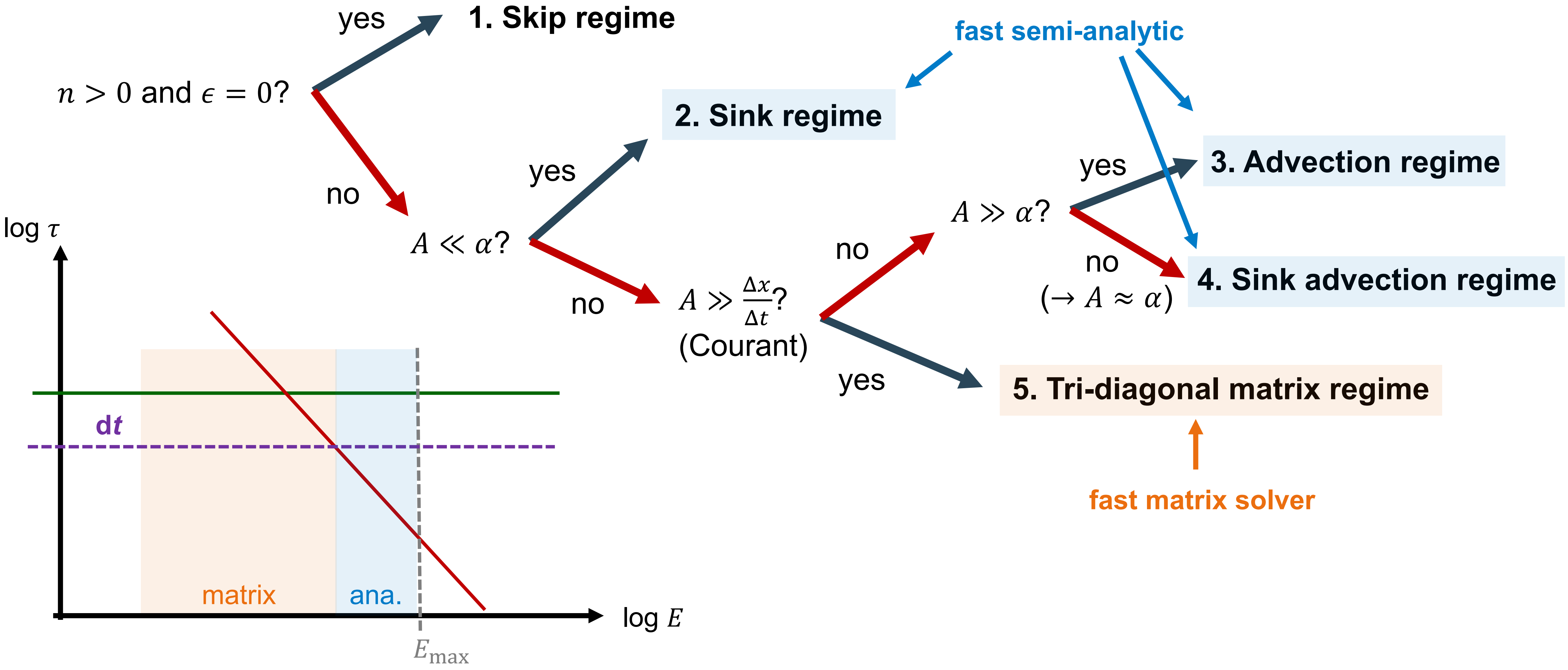
yes → **3. Advection regime**

no ($\rightarrow A \approx \alpha$) → **4. Sink advection regime**

yes → **5. Tri-diagonal matrix regime**

fast matrix solver

Evolve the particle densities



$n > 0$ and $\epsilon = 0$?

yes

1. Skip regime

no

$A \ll \alpha$?

yes

2. Sink regime

no

$A \gg \frac{\Delta x}{\Delta t}$?
(Courant)

no

$A \gg \alpha$?

yes

3. Advection regime

no
$(\rightarrow A \approx \alpha)$

4. Sink advection regime

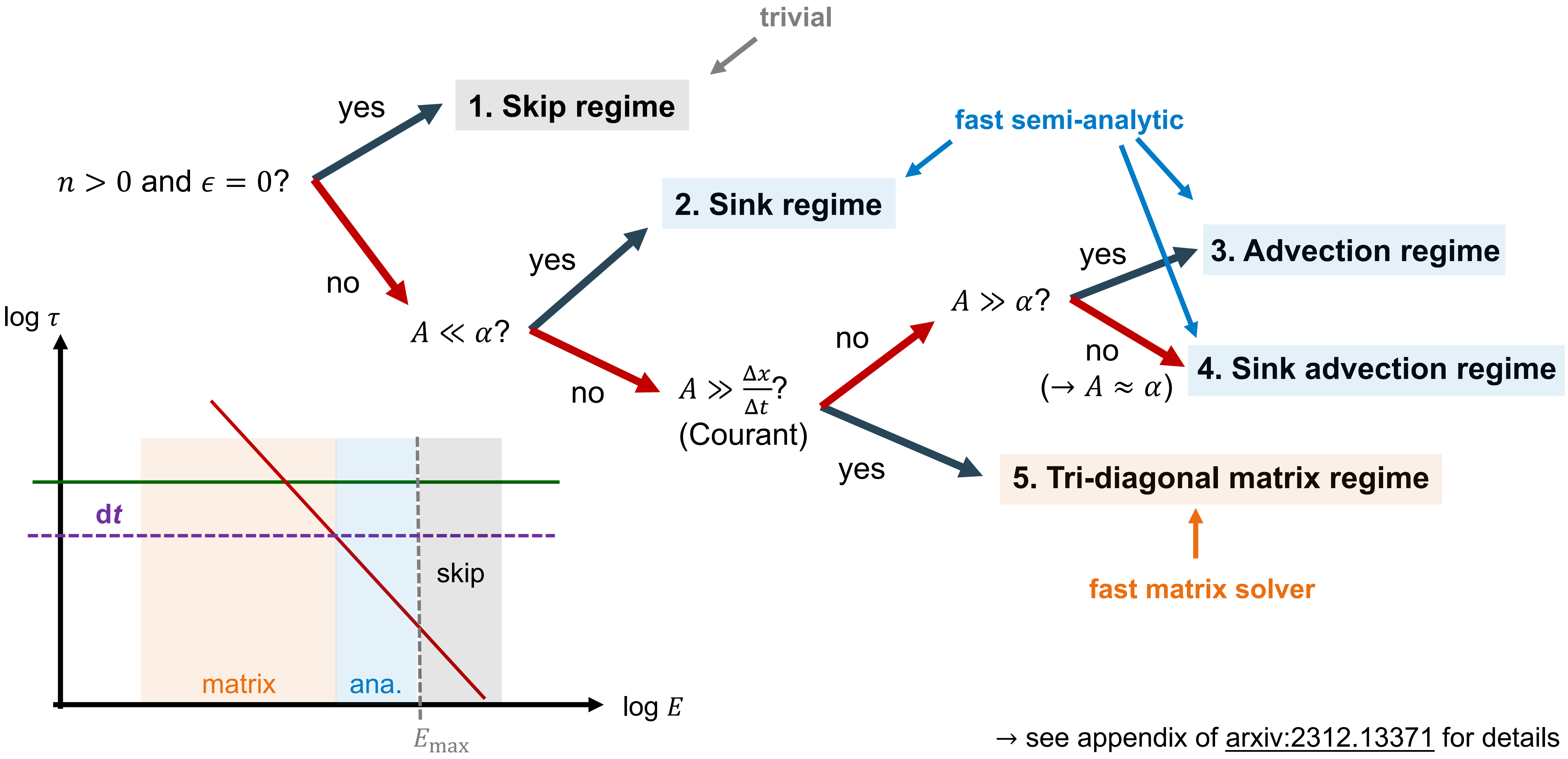
yes

5. Tri-diagonal matrix regime

fast semi-analytic

fast matrix solver

Evolve the particle densities



trivial

$n > 0$ and $\epsilon = 0$?

yes → **1. Skip regime**

no → $A \ll \alpha$?

yes → **2. Sink regime**

no → $A \gg \frac{\Delta x}{\Delta t}$? (Courant)

no → $A \gg \alpha$?

yes → **3. Advection regime**

no ($\to A \approx \alpha$) → **4. Sink advection regime**

yes → **5. Tri-diagonal matrix regime**

fast semi-analytic

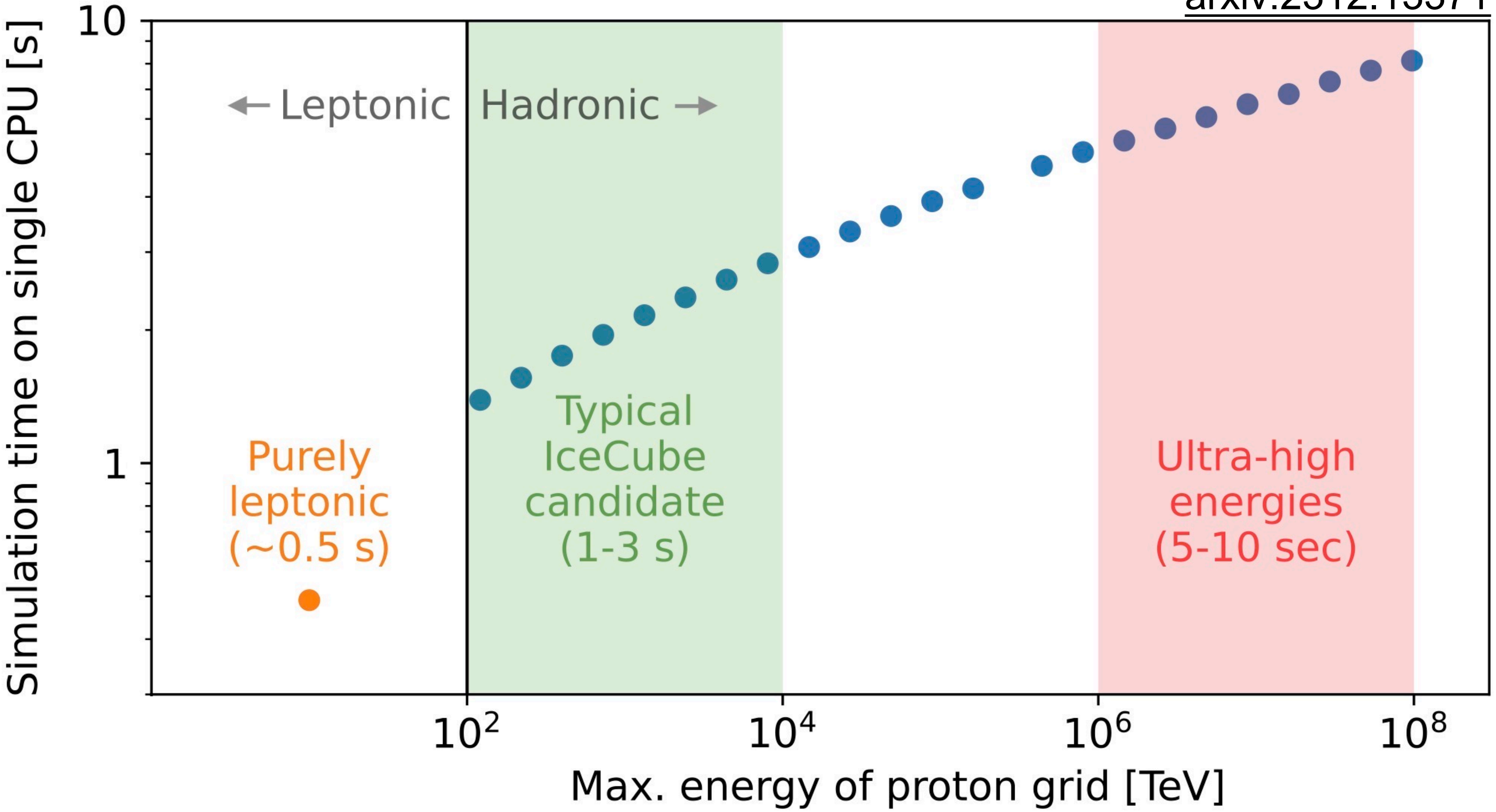
fast matrix solver

→ see appendix of arxiv:2312.13371 for details

Fast

- steady state in 1-10 seconds

arxiv:2312.13371



Fast

- steady state in 1-10 seconds
- speed optimizations:

→ pre-calculated/tabulated/simplified kernels (cut to relevant energy ranges)

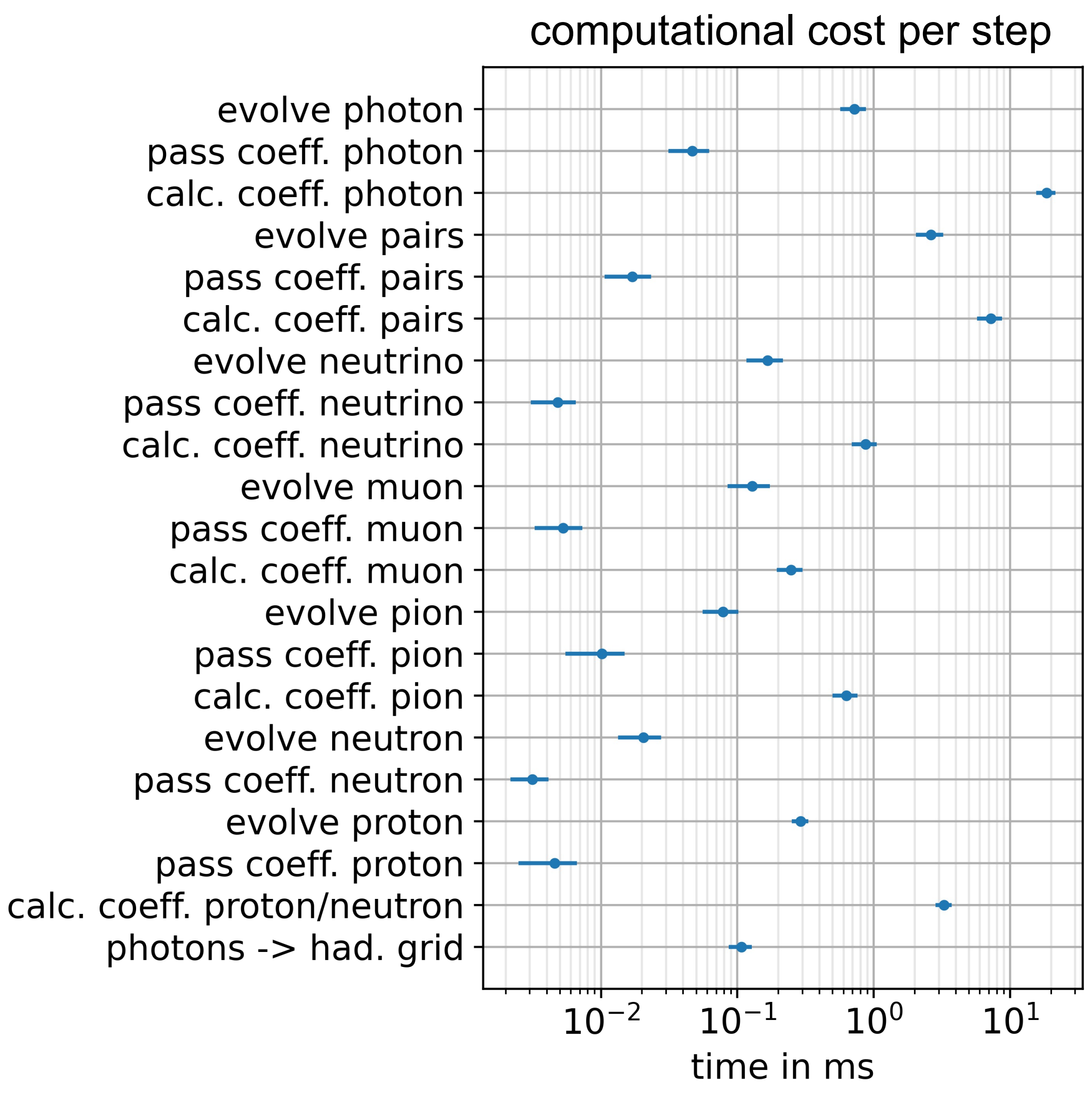
→ ~40 switches allow to select for relevant processes

→ monitor computational cost

→ speed optimized solver

→ compiled (C++)

→ adjustable energy grid



Trackable

- possibility to co-evolve components to track contributions

  → which components dominate photon spectra at which energies?

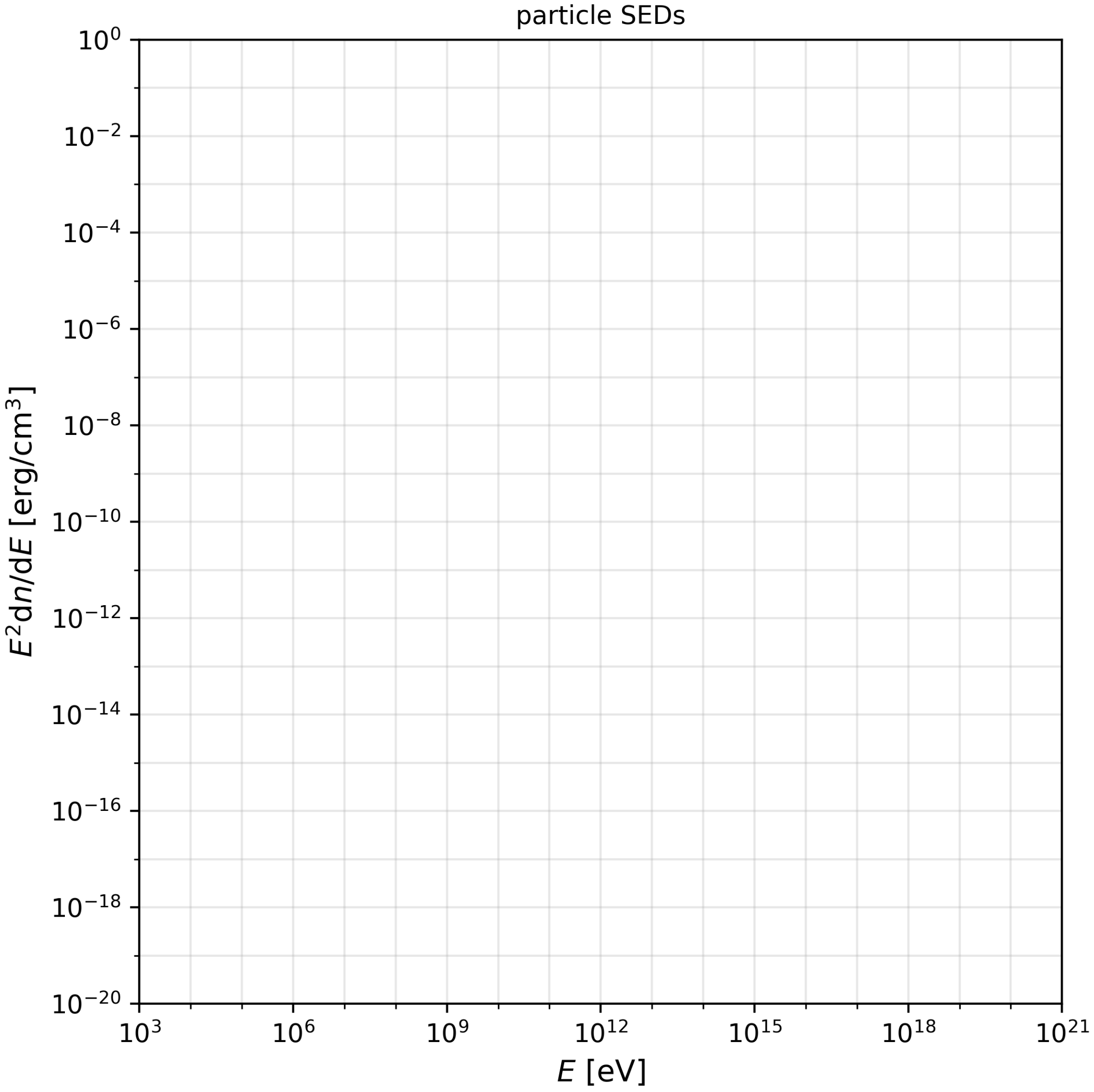
  → which neutrinos come from $pp/p\gamma$?

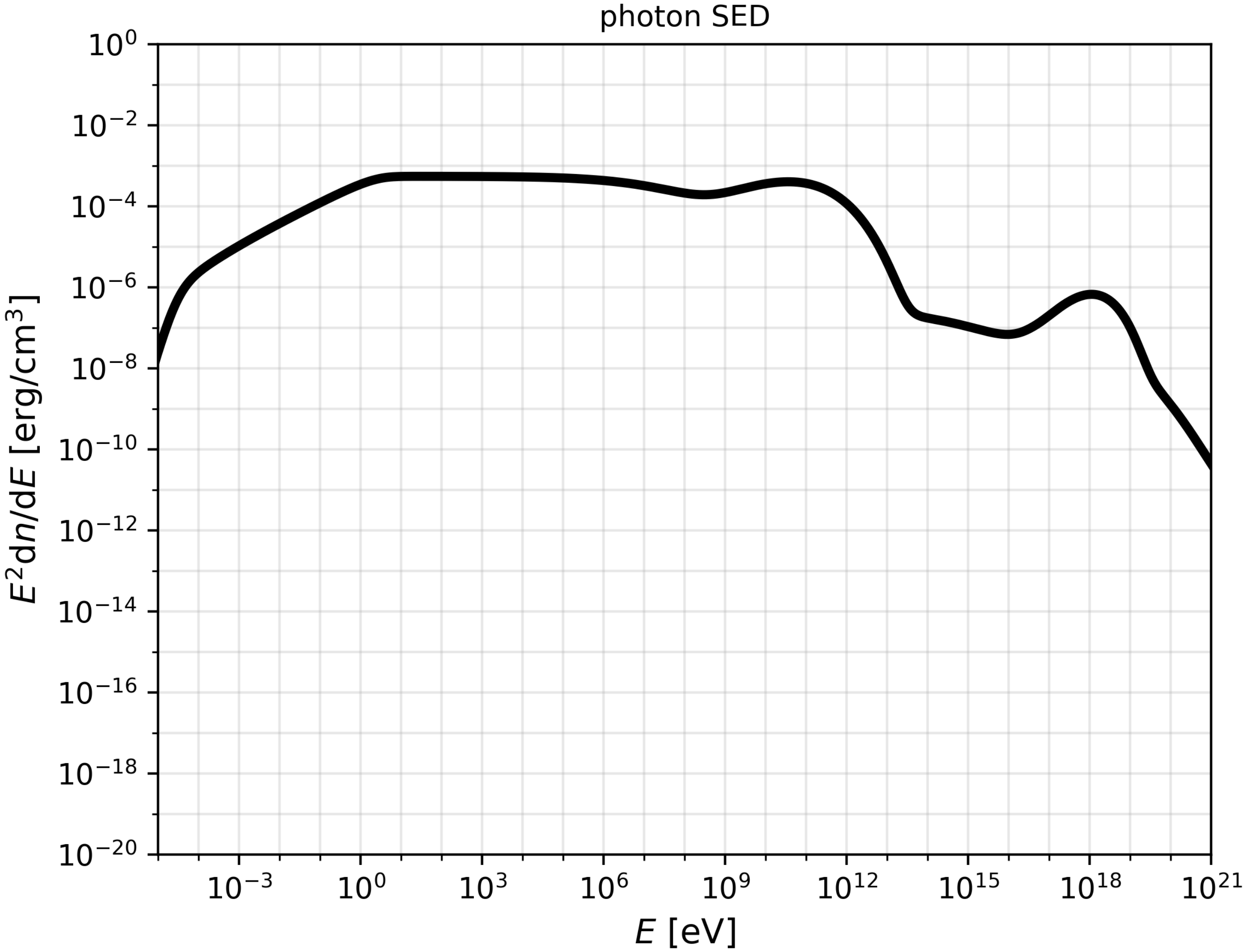
  → which processes contribute how much to electrons/positrons?

- no real slow down

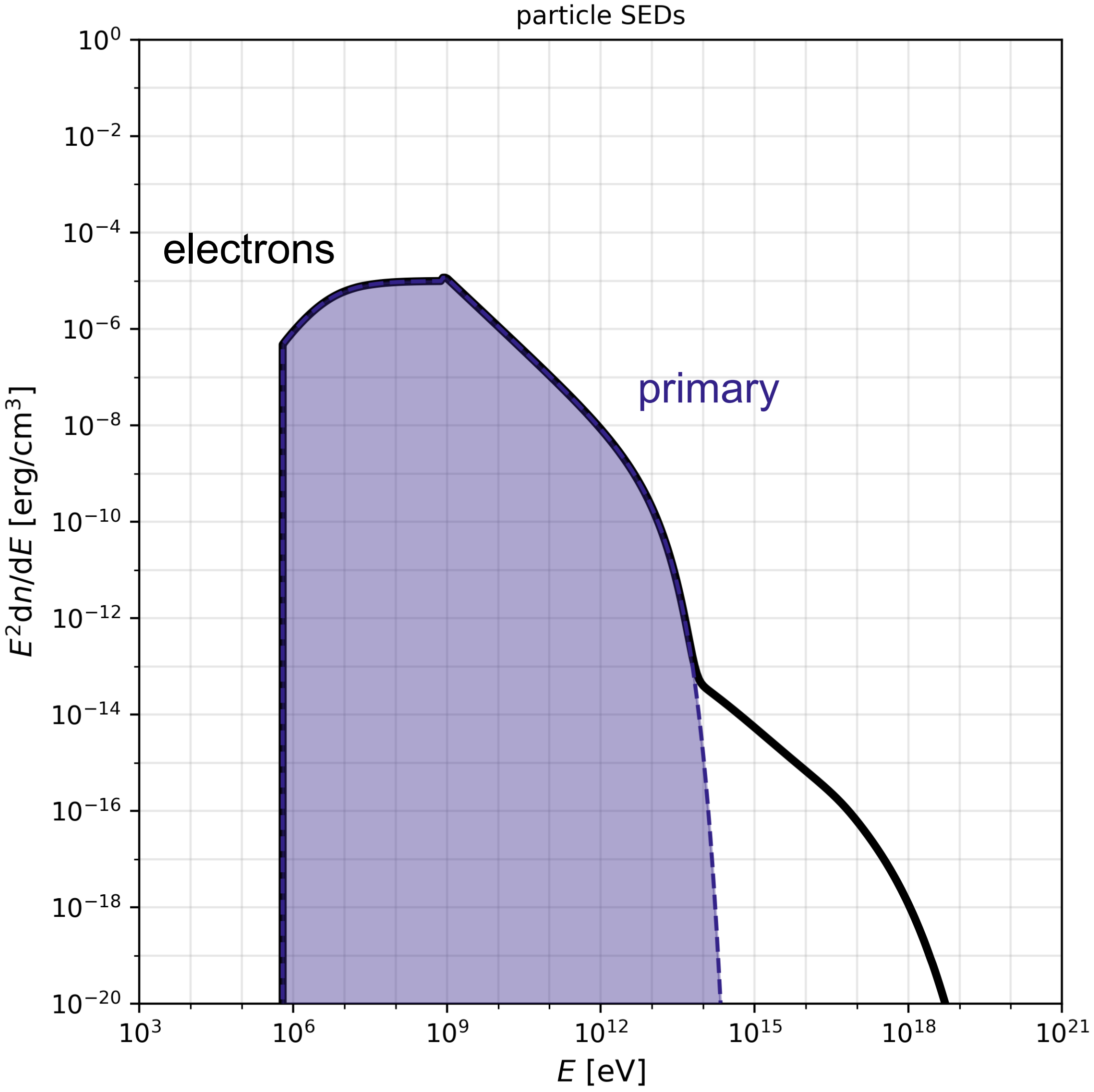
- great intuition!

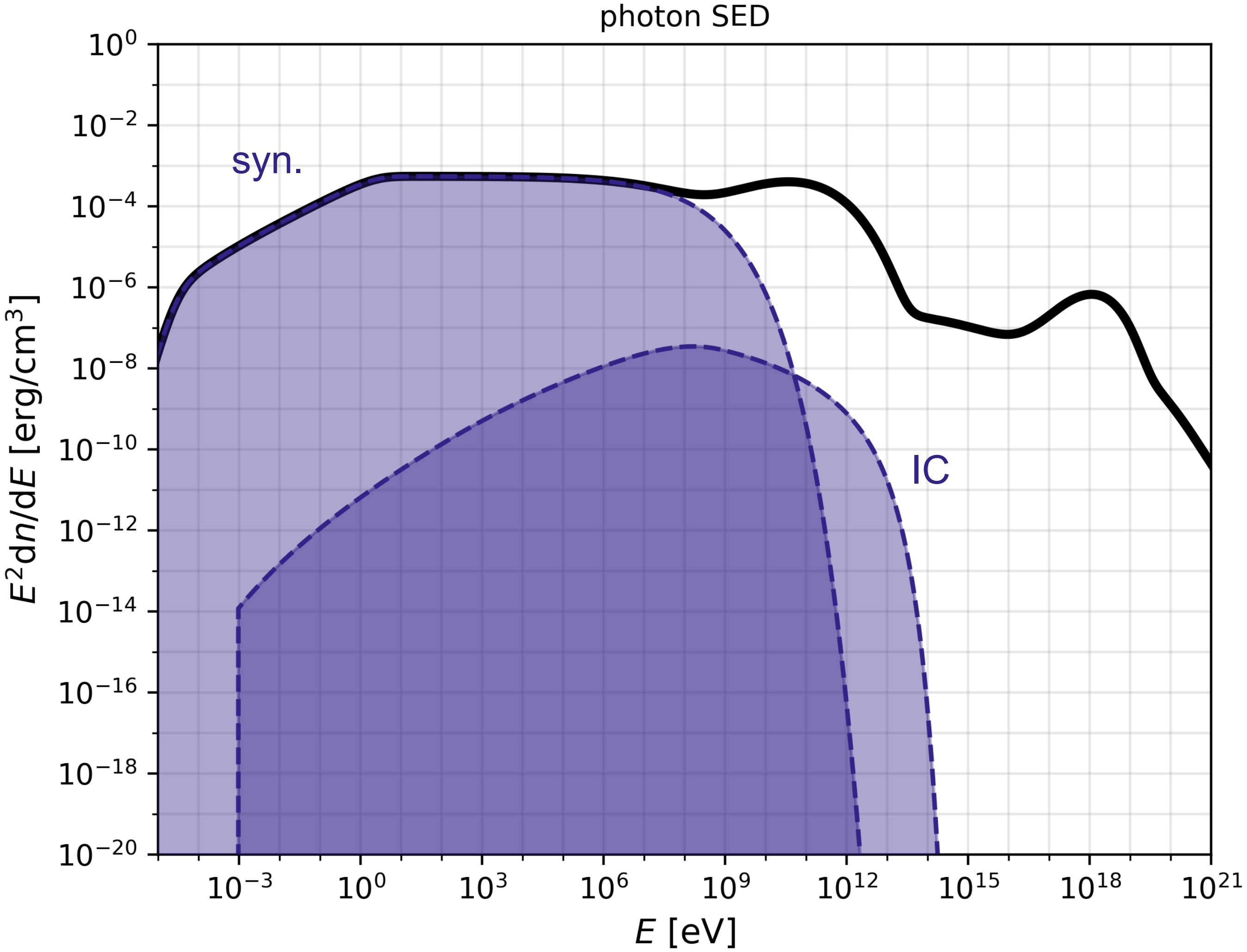
Trackable


particle SEDs
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
$E^2 dn/dE$ [erg/cm^3]
10^{3}
10^{6}
10^{9}
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

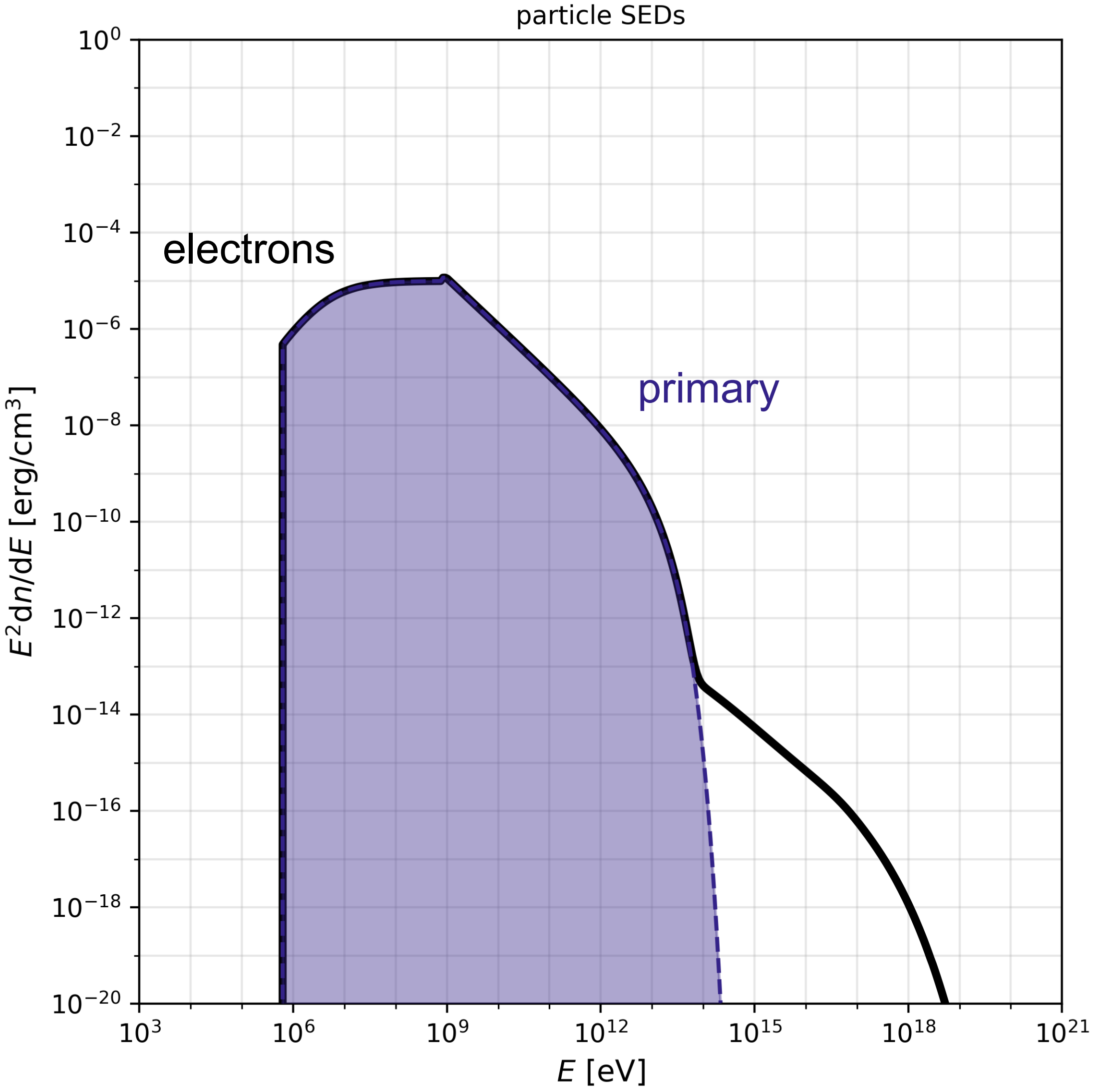

photon SED
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
$E^2 dn/dE$ [erg/cm^3]
10^{-3}
10^{0}
10^{3}
10^{6}
10^{9}
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

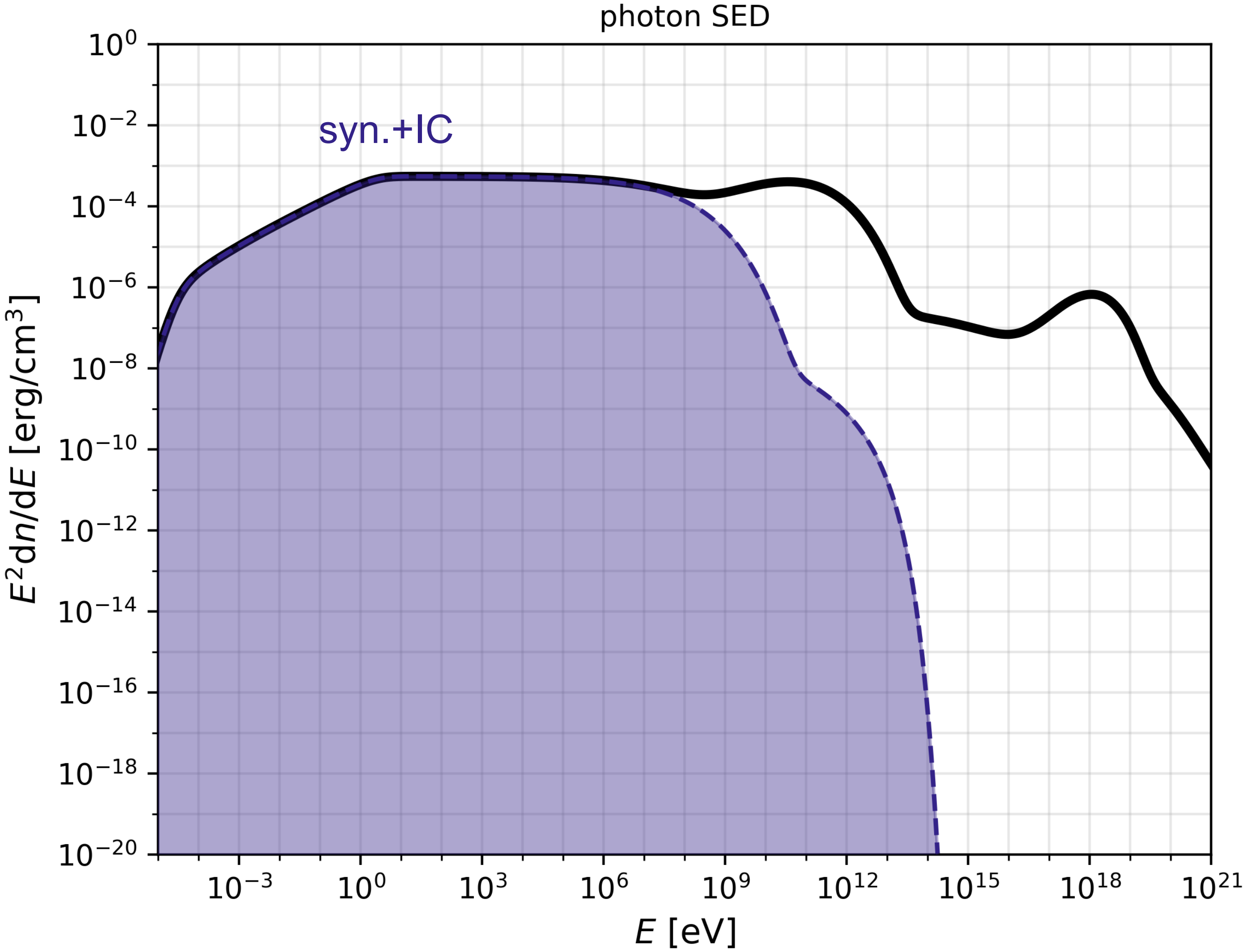
Trackable


particle SEDs
electrons
primary
$E^2 \mathrm{d}n/\mathrm{d}E$ [erg/cm^3]
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^{3}
10^{6}
10^{9}
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

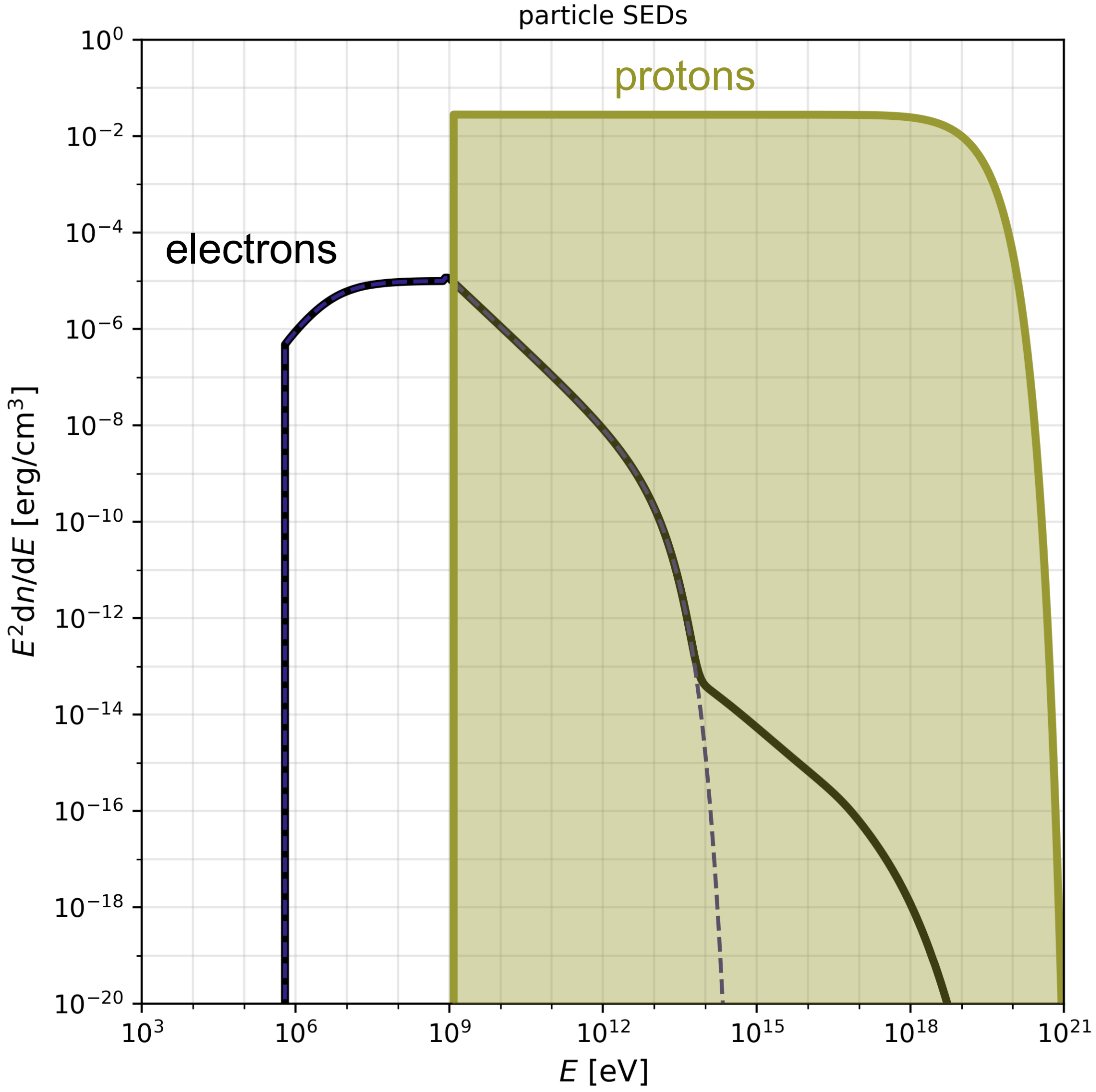

photon SED
syn.
IC
$E^2 \mathrm{d}n/\mathrm{d}E$ [erg/cm^3]
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^{-3}
10^{0}
10^{3}
10^{6}
10^{9}
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

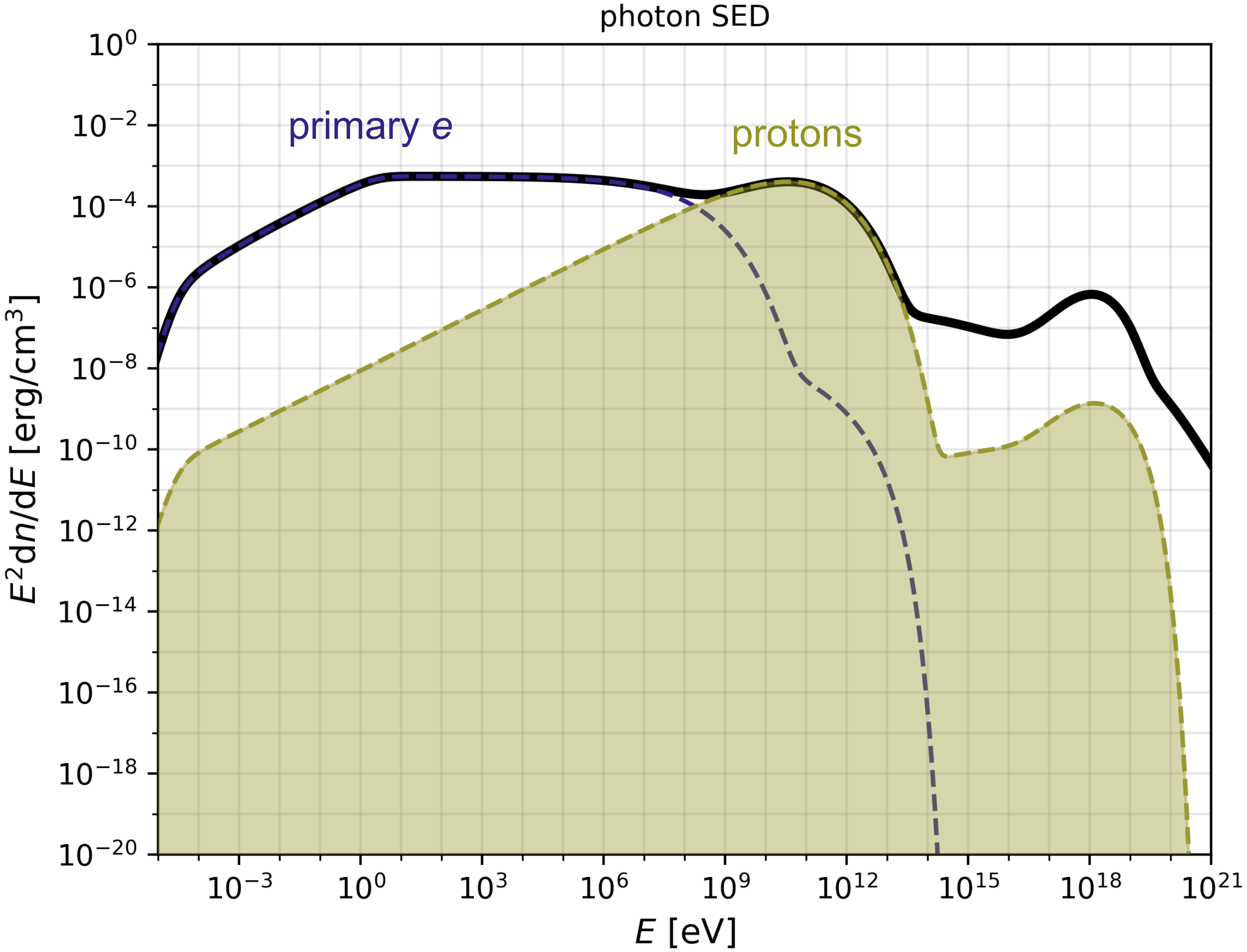
Trackable


particle SEDs
electrons
primary
$E^2 dn/dE$ [erg/cm^3]
E [eV]
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^3
10^6
10^9
10^{12}
10^{15}
10^{18}
10^{21}

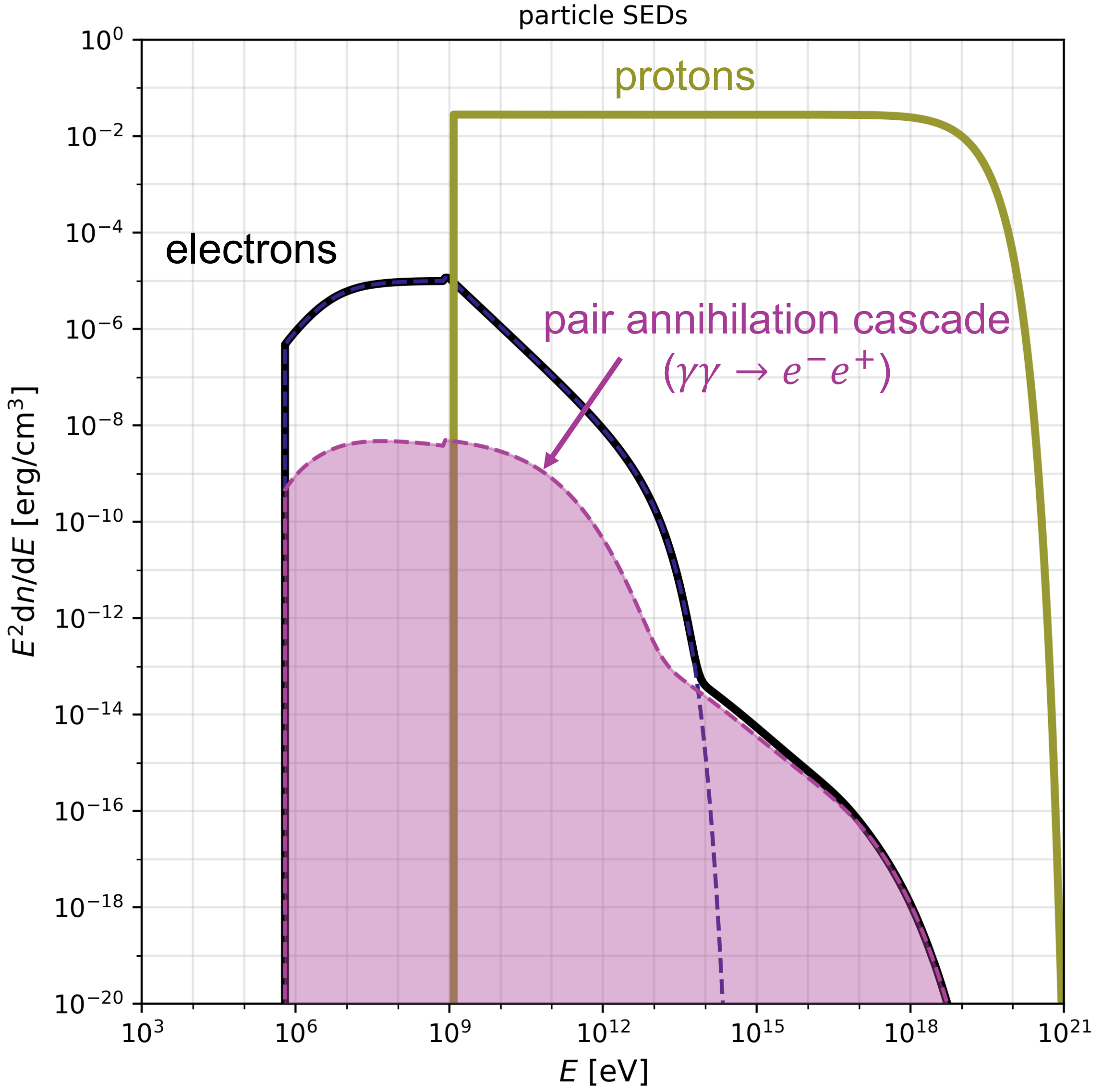

photon SED
syn.+IC
$E^2 dn/dE$ [erg/cm^3]
E [eV]
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^{-3}
10^0
10^3
10^6
10^9
10^{12}
10^{15}
10^{18}
10^{21}

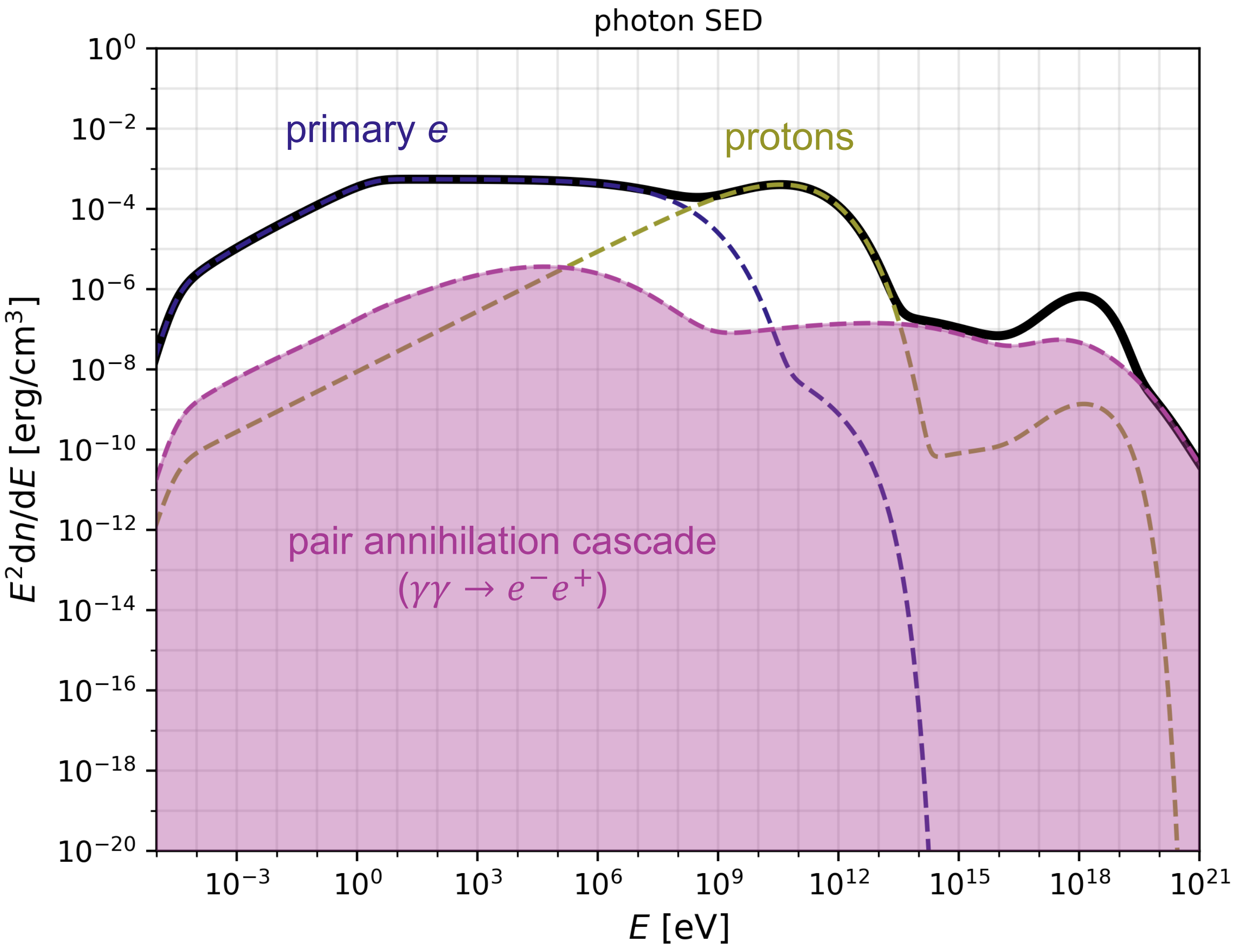
Trackable


particle SEDs
protons
electrons
$E^2 dn/dE$ [erg/cm^3]
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^3
10^6
10^9
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

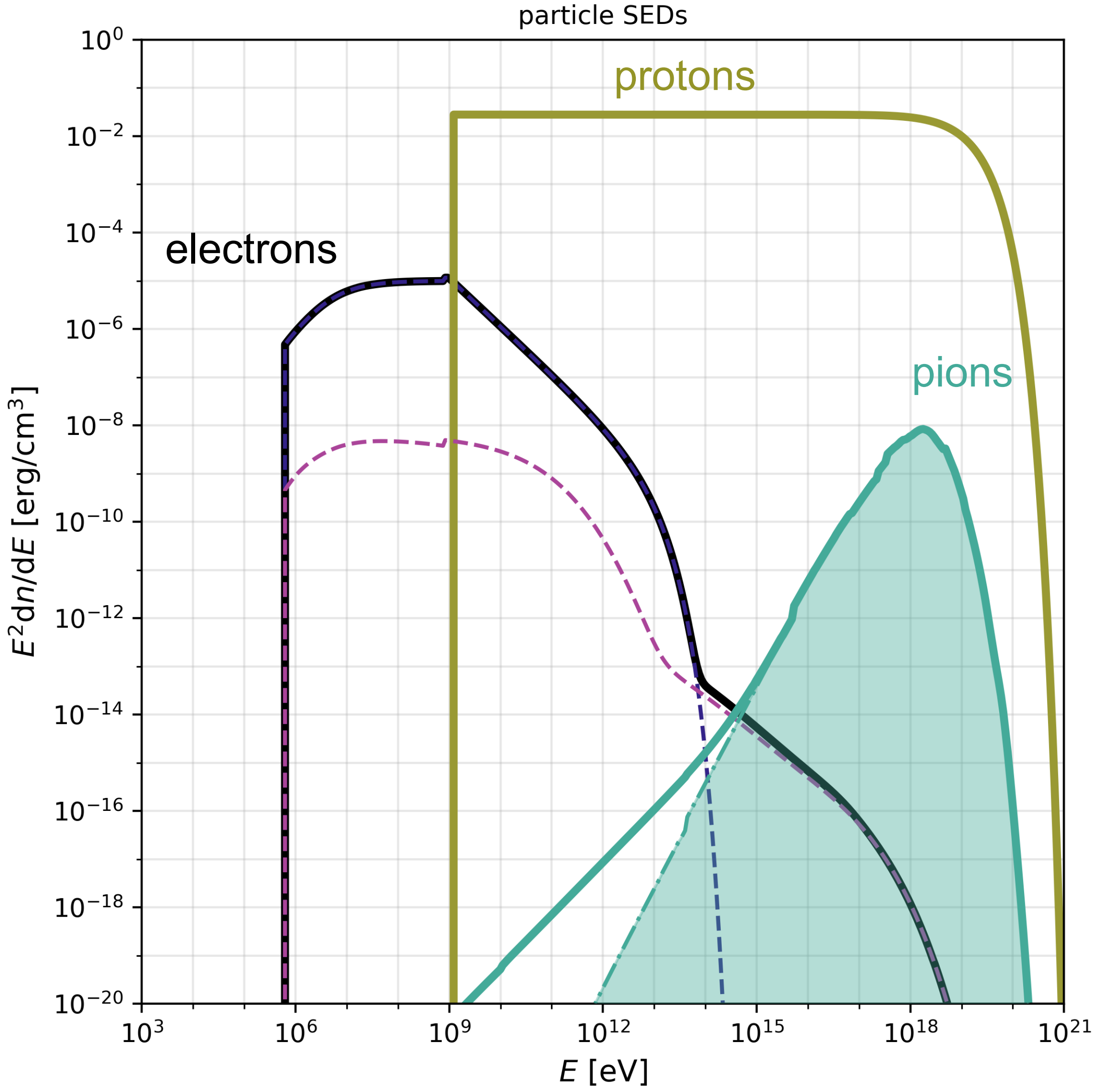

photon SED
primary e
protons
$E^2 dn/dE$ [erg/cm^3]
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
10^{-3}
10^0
10^3
10^6
10^9
10^{12}
10^{15}
10^{18}
10^{21}
E [eV]

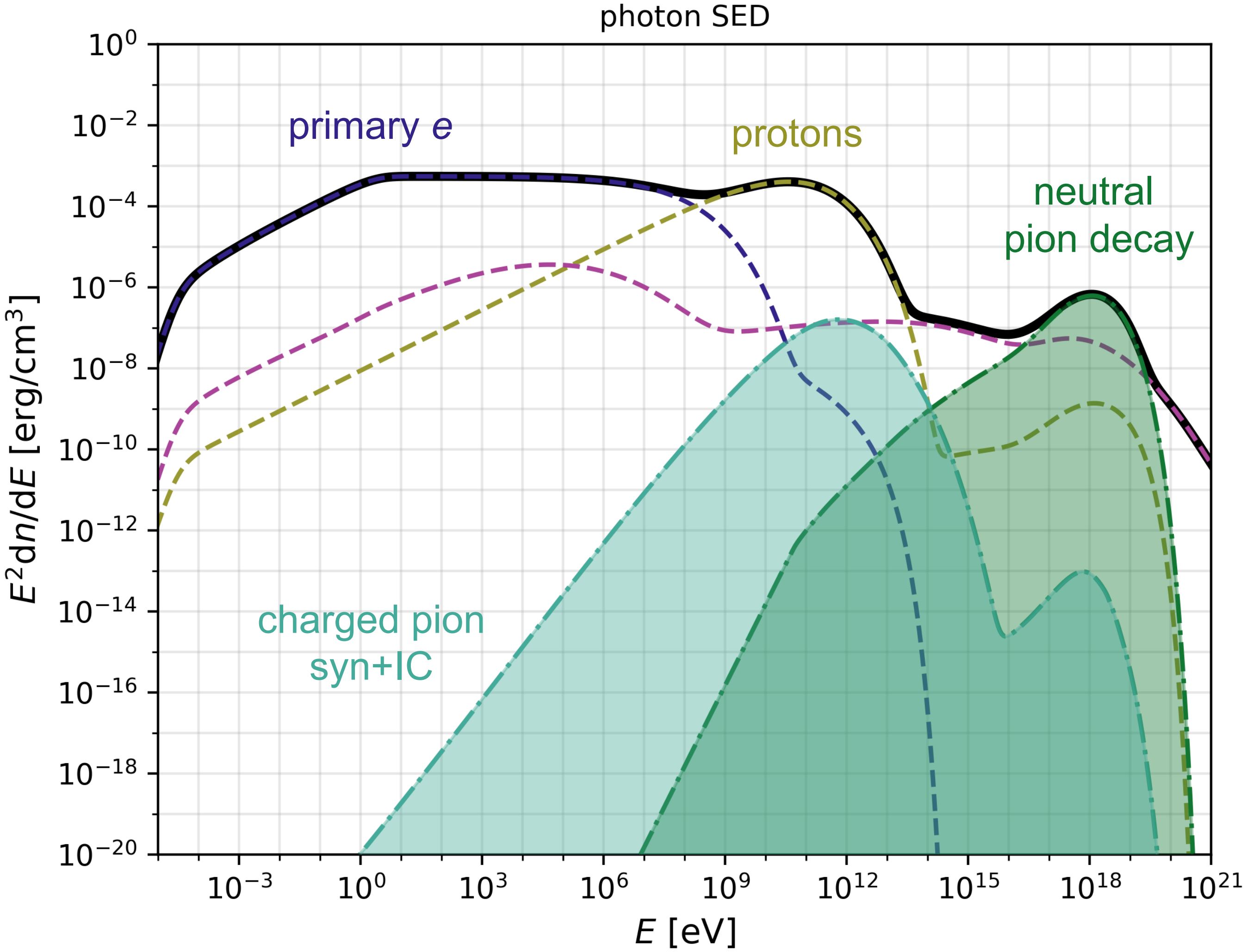
Trackable – pair annihilation



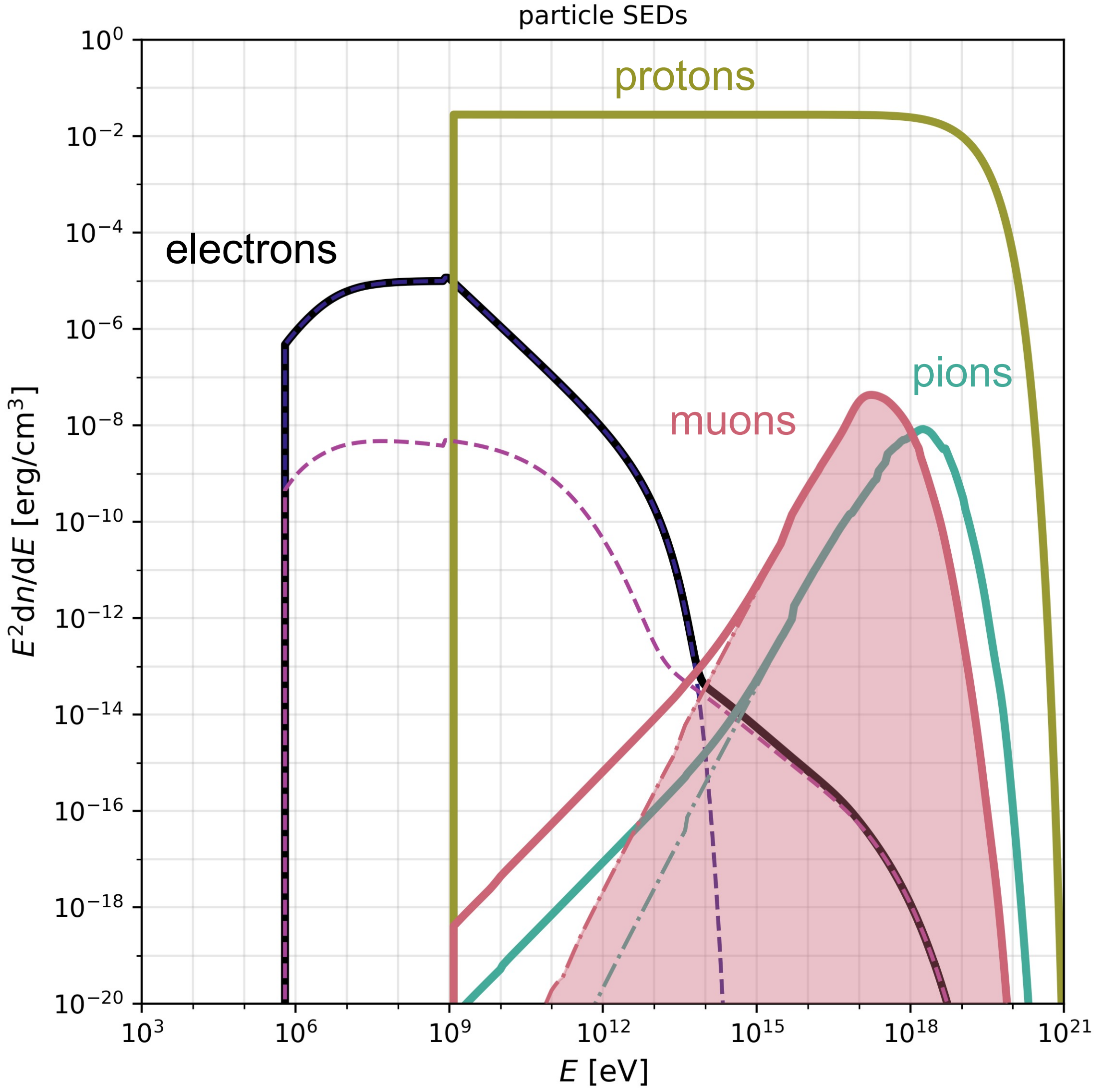


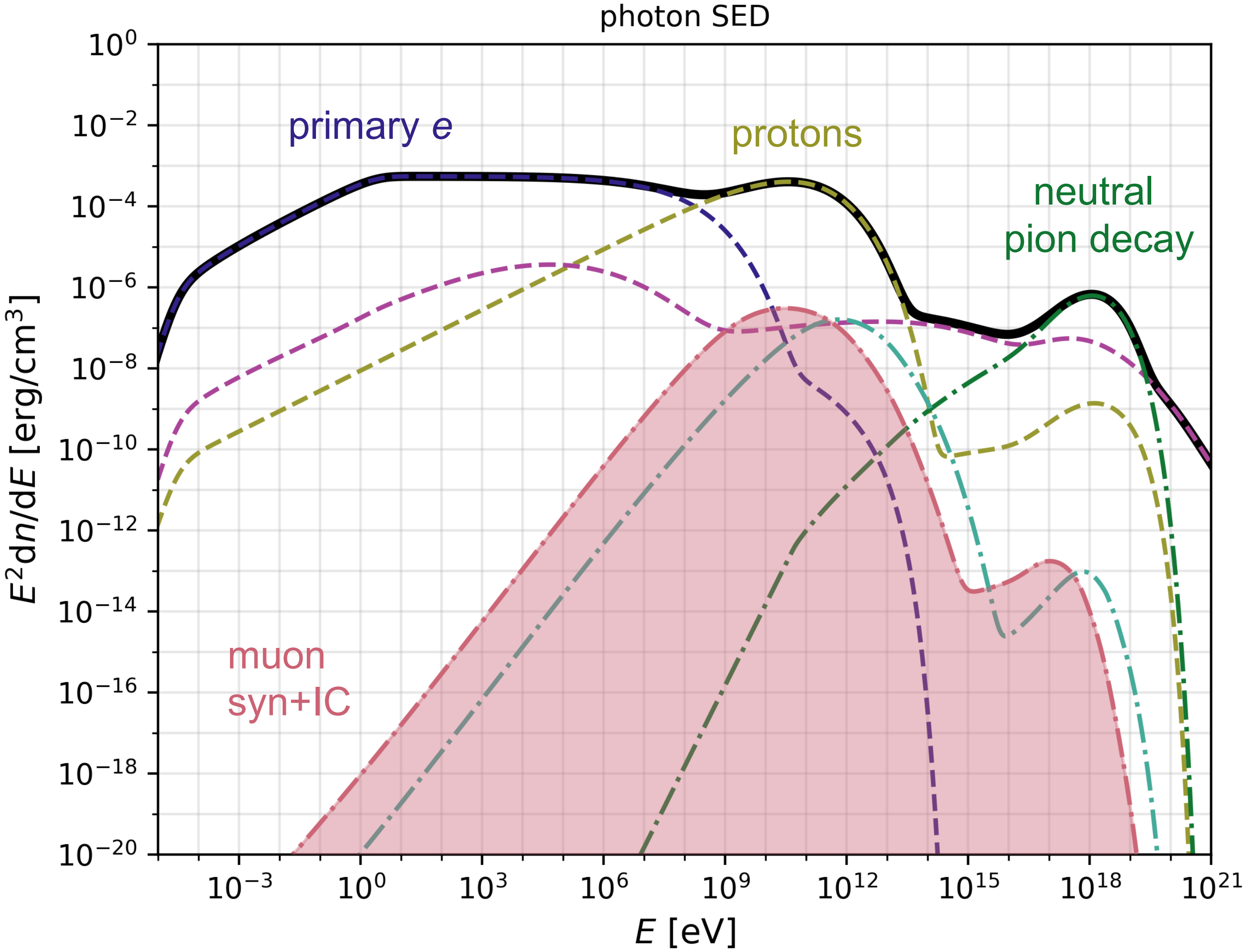
Trackable – photo-pion cascade: $p\gamma \rightarrow \pi$



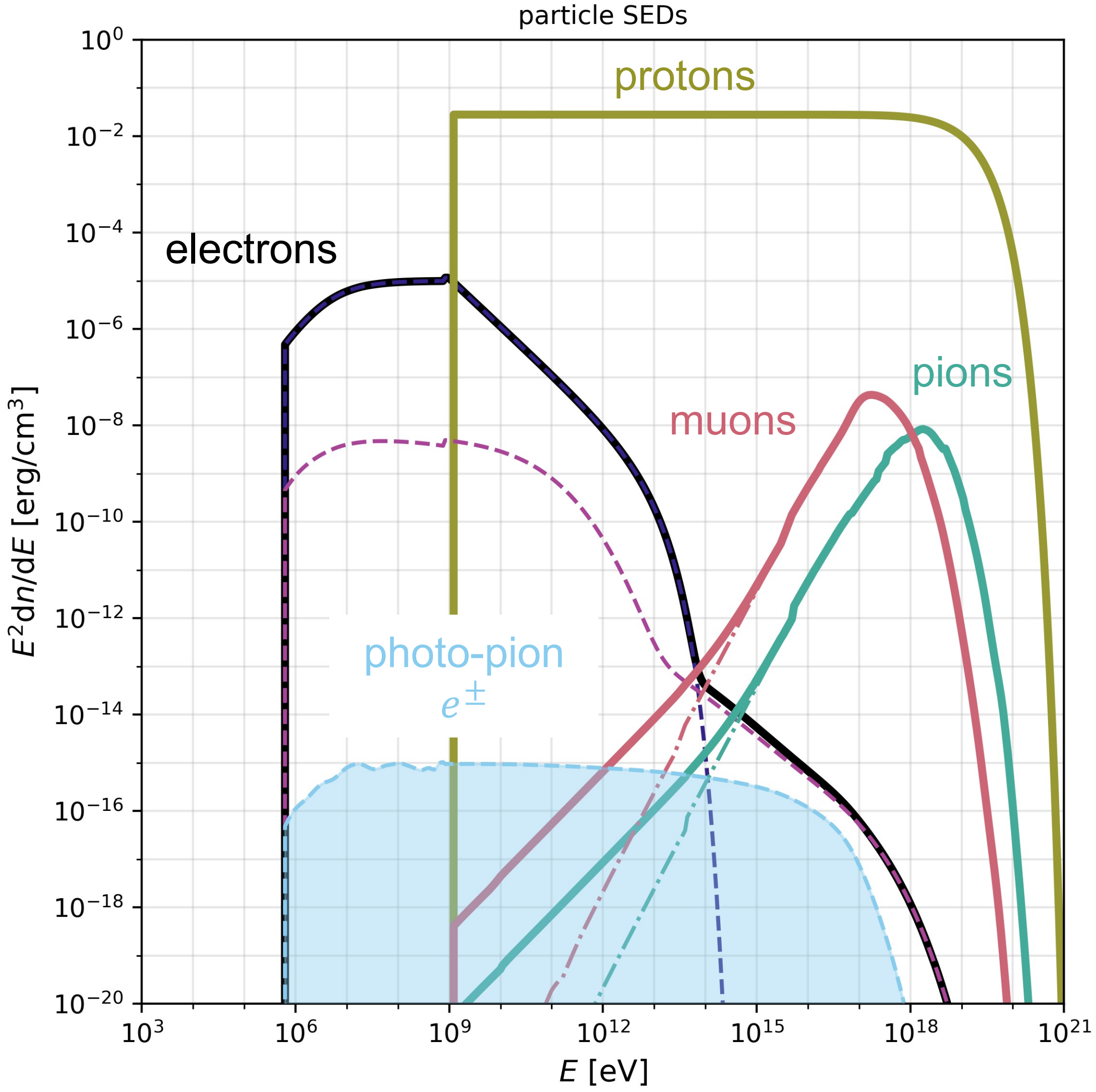


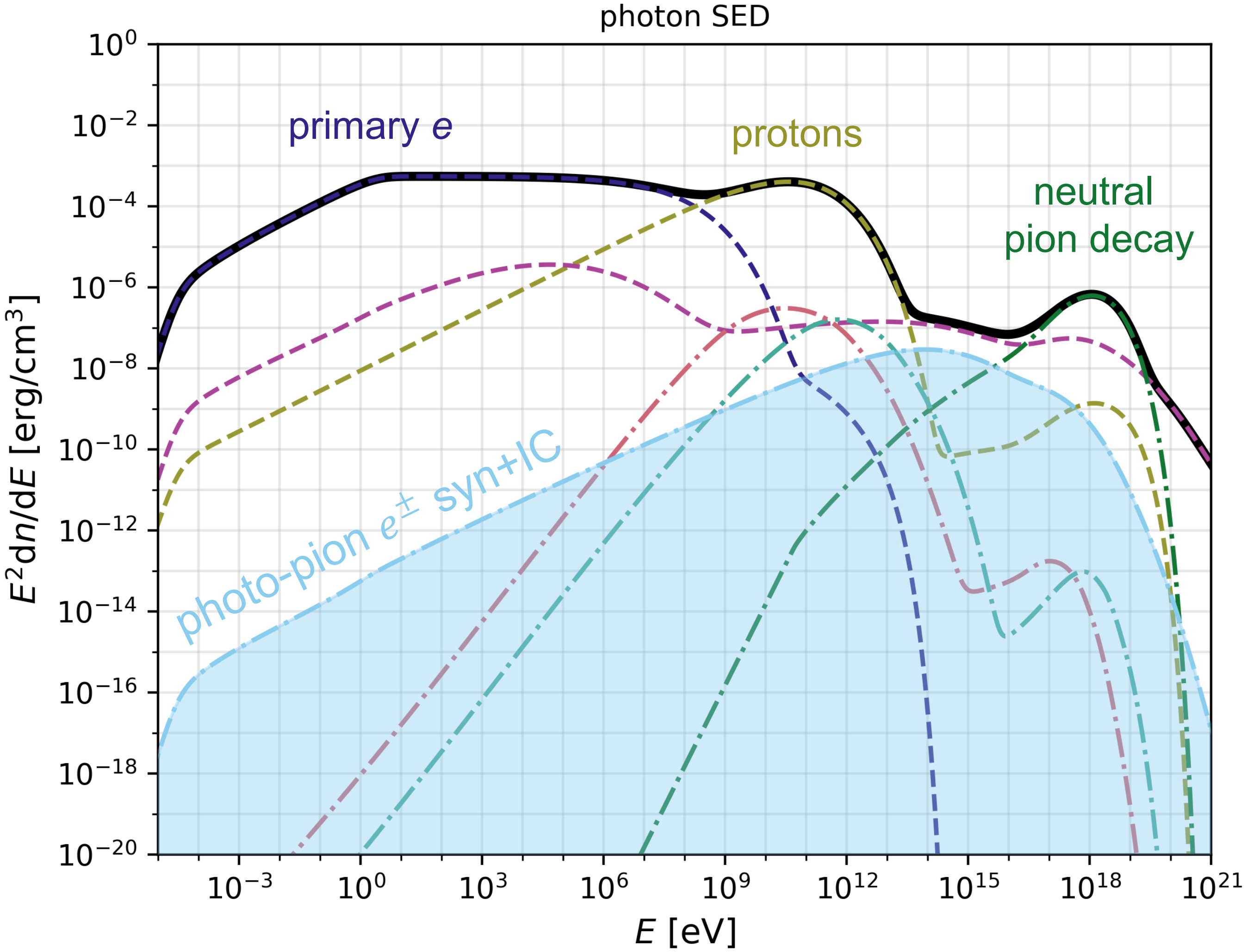
Trackable – photo-pion cascade: $p\gamma \rightarrow \pi \rightarrow \mu$



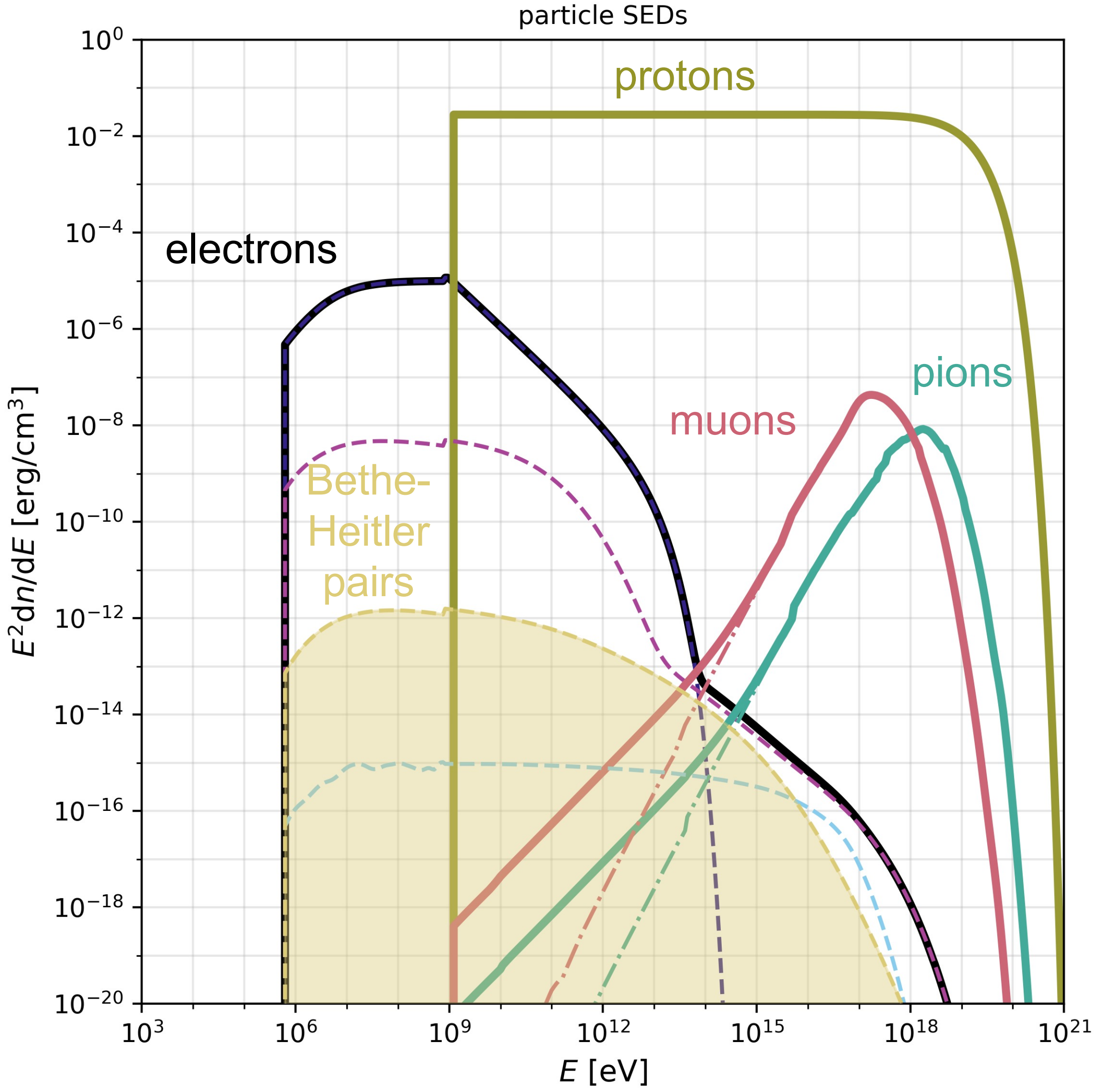


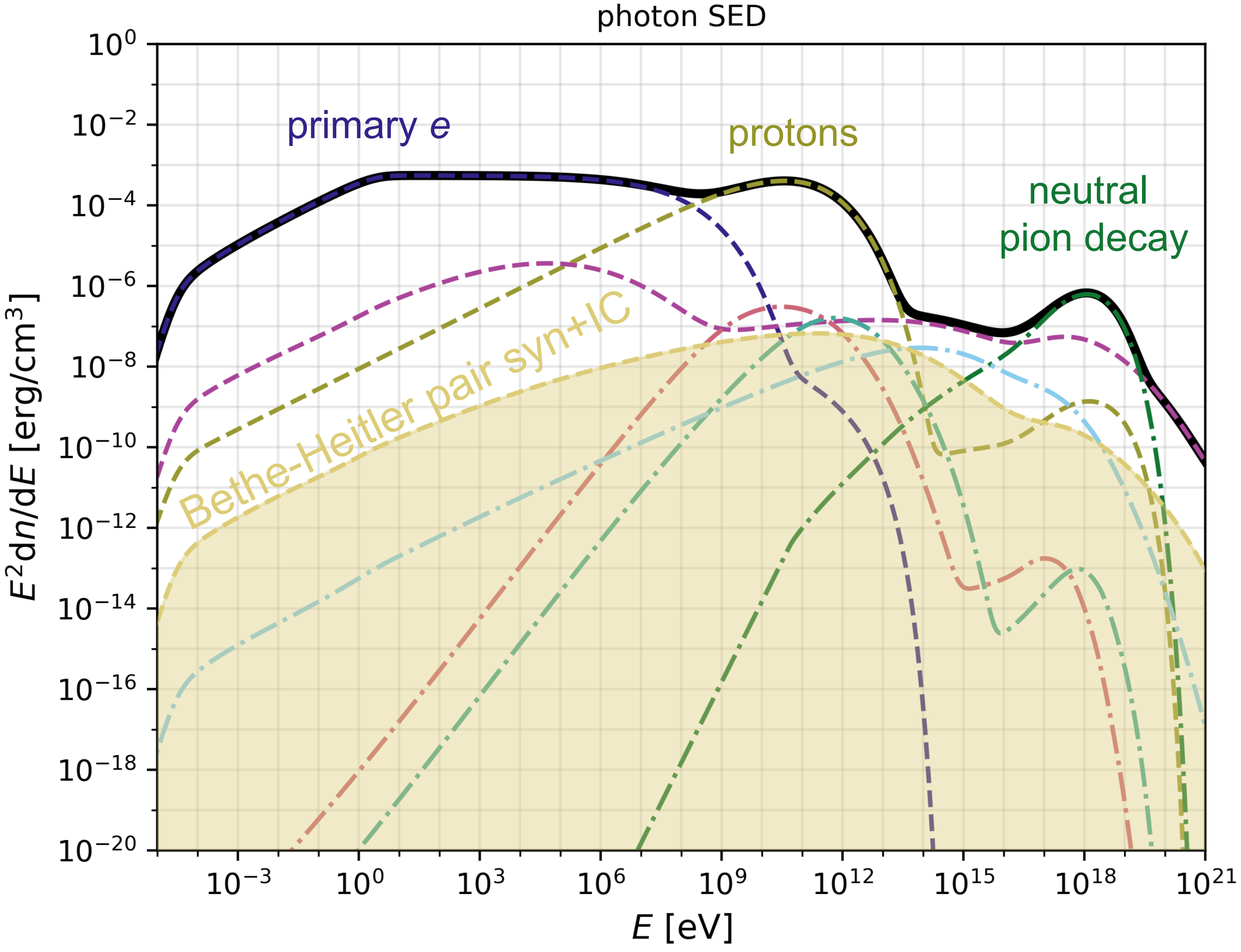
Trackable – photo-pion cascade: $p\gamma \rightarrow \pi \rightarrow \mu \rightarrow e^{\pm}$



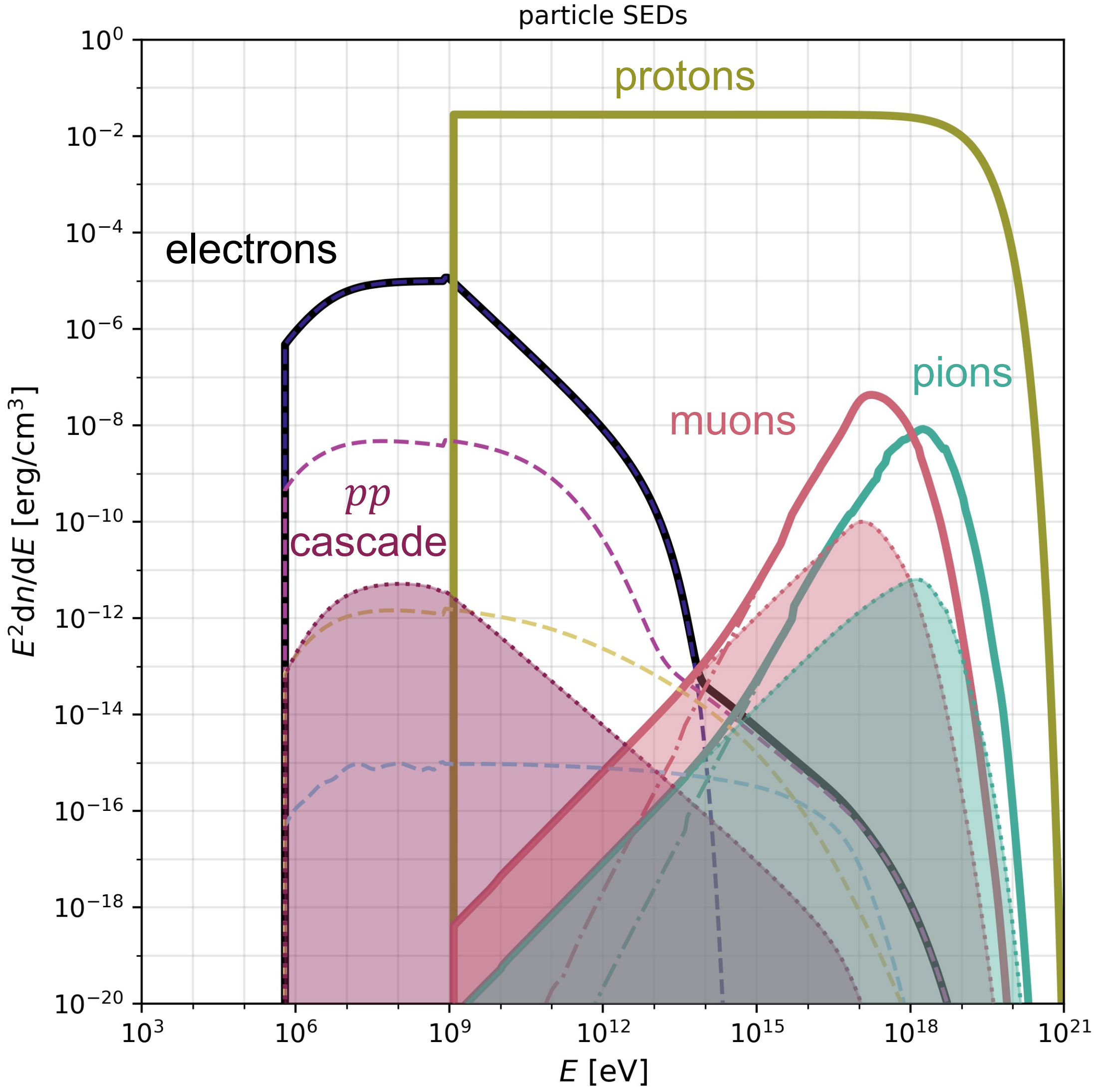


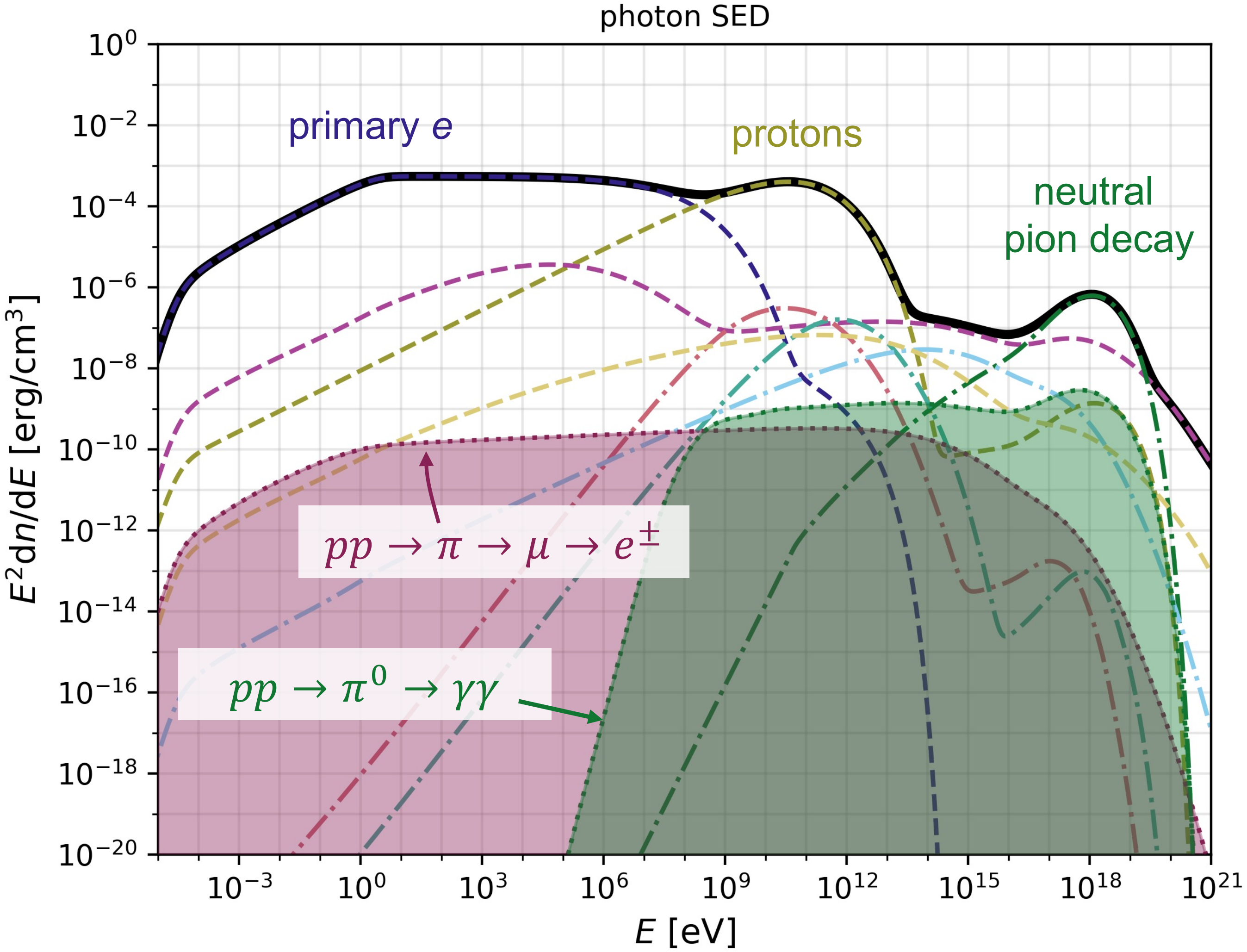
Trackable – Bethe-Heitler : $p\gamma \rightarrow pe^+e^-$



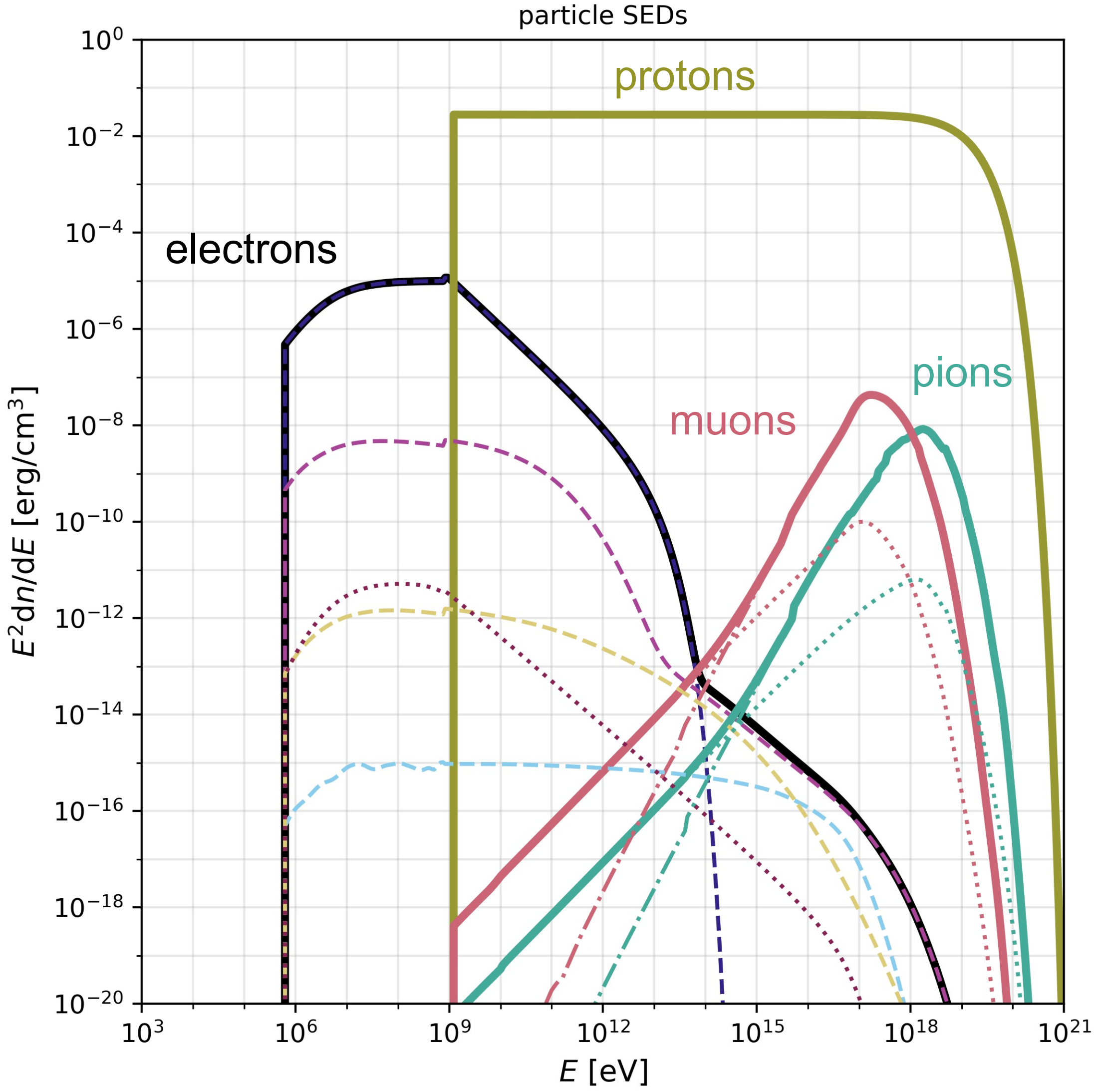


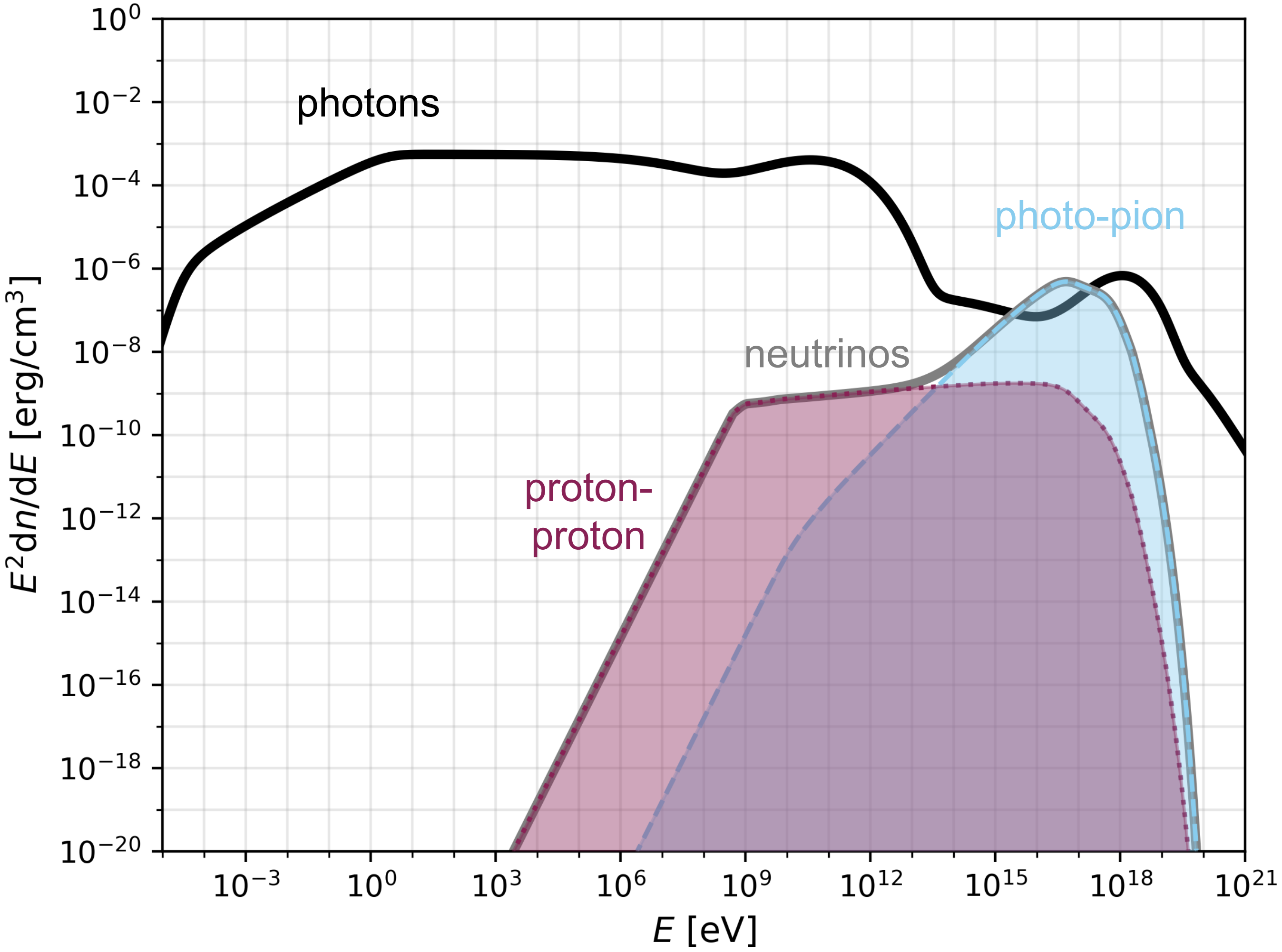
Trackable – proton-proton : $pp \rightarrow \pi \rightarrow \mu \rightarrow e^{\pm}$



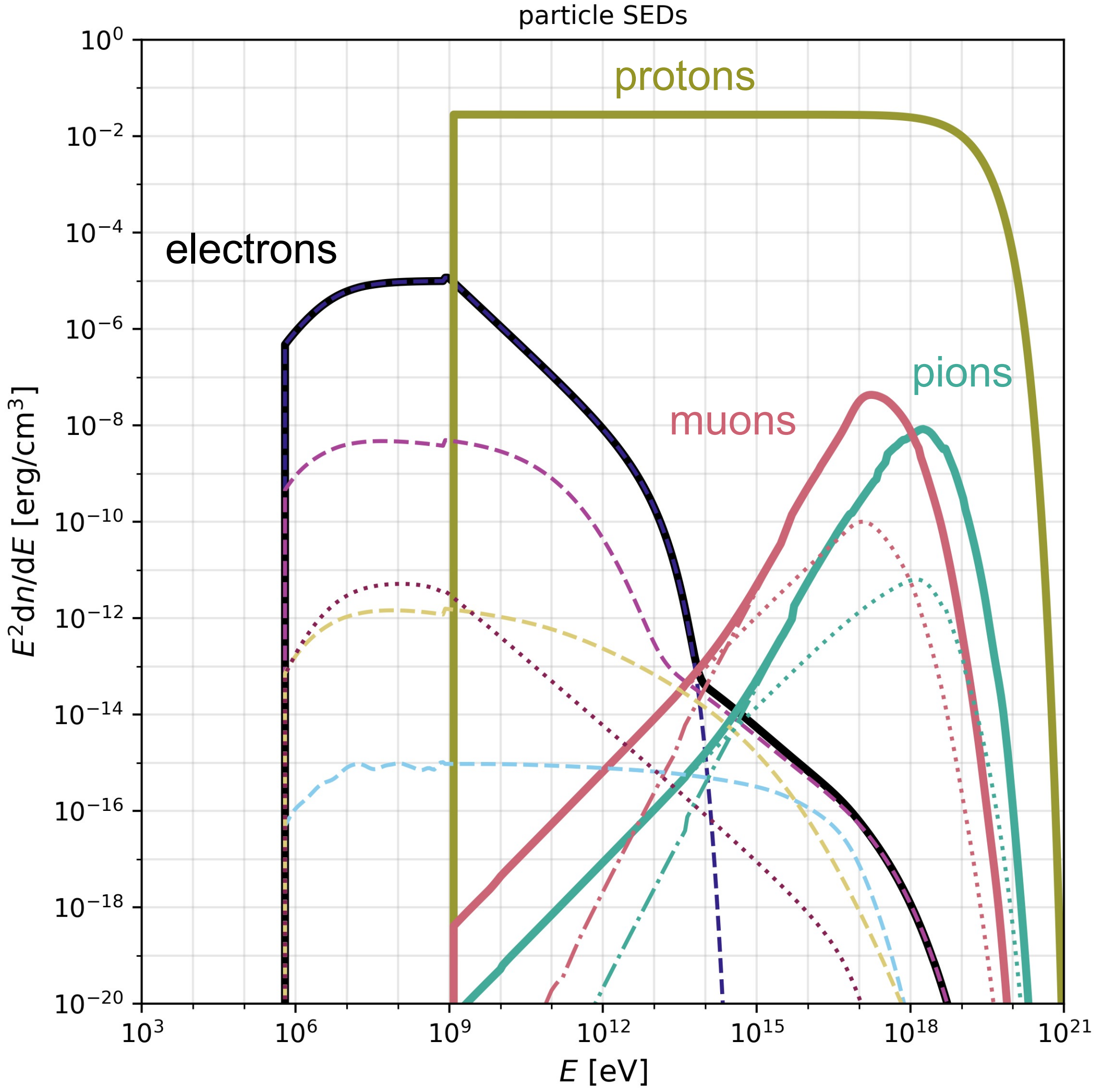


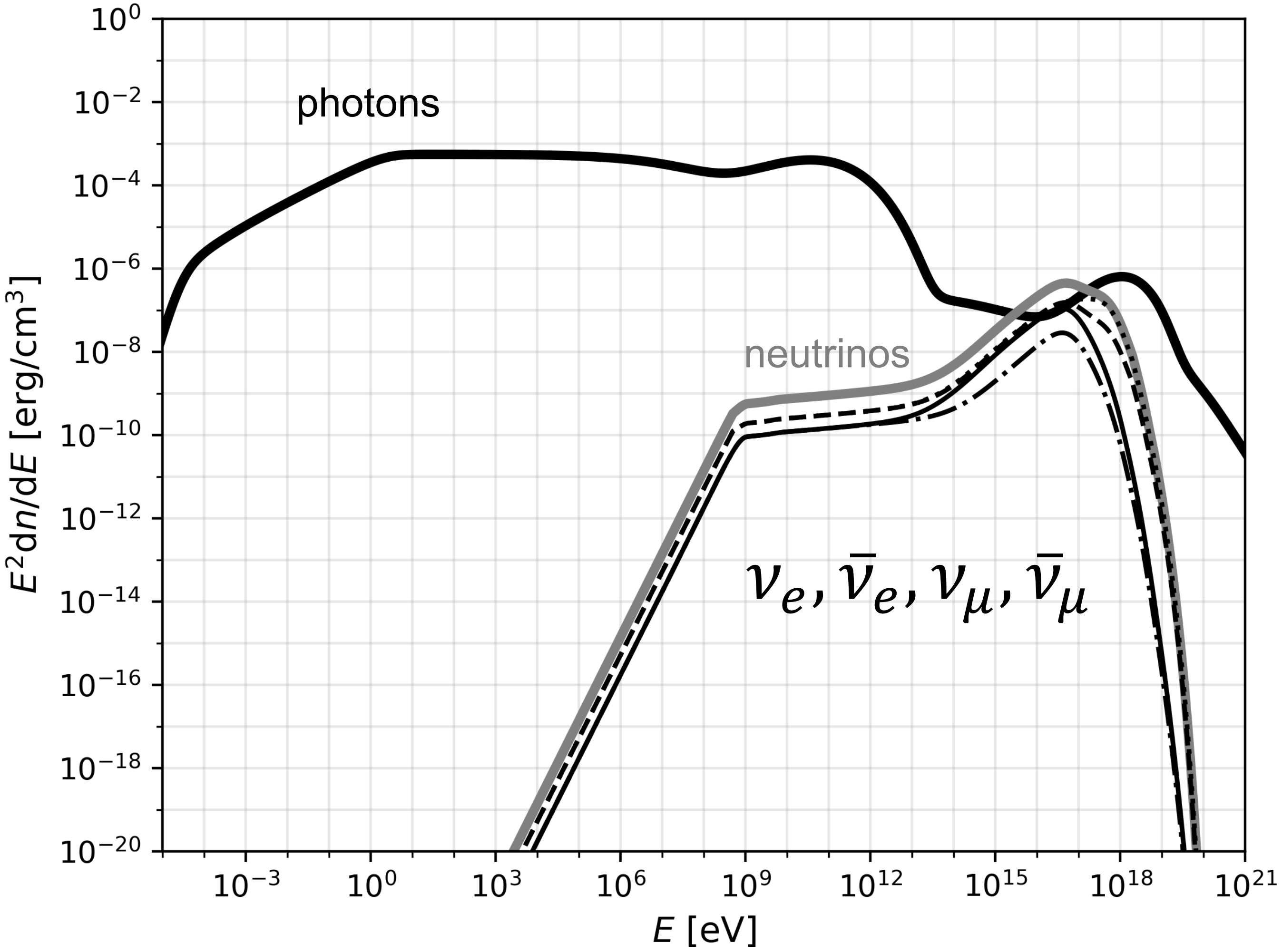
Trackable – neutrinos





Trackable – neutrino flavours

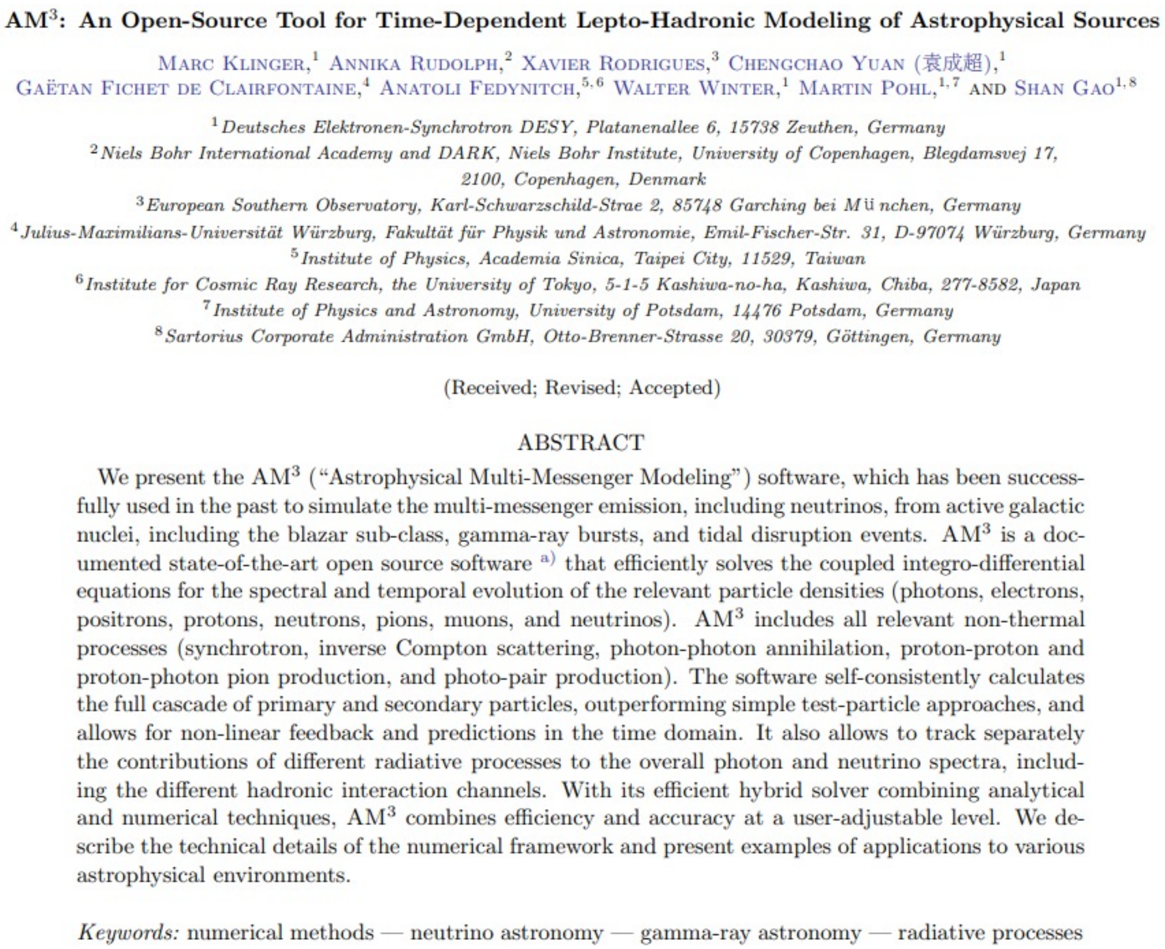




Now public!

- paper with detailed summary of processes, solver, etc…

arxiv:2312.13371



AM^3: An Open-Source Tool for Time-Dependent Lepto-Hadronic Modeling of Astrophysical Sources

Marc Klinger,[1] Annika Rudolph,[2] Xavier Rodrigues,[3] Chengchao Yuan (袁成超),[1]
Gaëtan Fichet de Clairfontaine,[4] Anatoli Fedynitch,[5,6] Walter Winter,[1] Martin Pohl,[1,7] and Shan Gao[1,8]

[1] *Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738 Zeuthen, Germany*
[2] *Niels Bohr International Academy and DARK, Niels Bohr Institute, University of Copenhagen, Blegdamsvej 17, 2100, Copenhagen, Denmark*
[3] *European Southern Observatory, Karl-Schwarzschild-Strae 2, 85748 Garching bei M ü nchen, Germany*
[4] *Julius-Maximilians-Universität Würzburg, Fakultät für Physik und Astronomie, Emil-Fischer-Str. 31, D-97074 Würzburg, Germany*
[5] *Institute of Physics, Academia Sinica, Taipei City, 11529, Taiwan*
[6] *Institute for Cosmic Ray Research, the University of Tokyo, 5-1-5 Kashiwa-no-ha, Kashiwa, Chiba, 277-8582, Japan*
[7] *Institute of Physics and Astronomy, University of Potsdam, 14476 Potsdam, Germany*
[8] *Sartorius Corporate Administration GmbH, Otto-Brenner-Strasse 20, 30379, Göttingen, Germany*

(Received; Revised; Accepted)

ABSTRACT

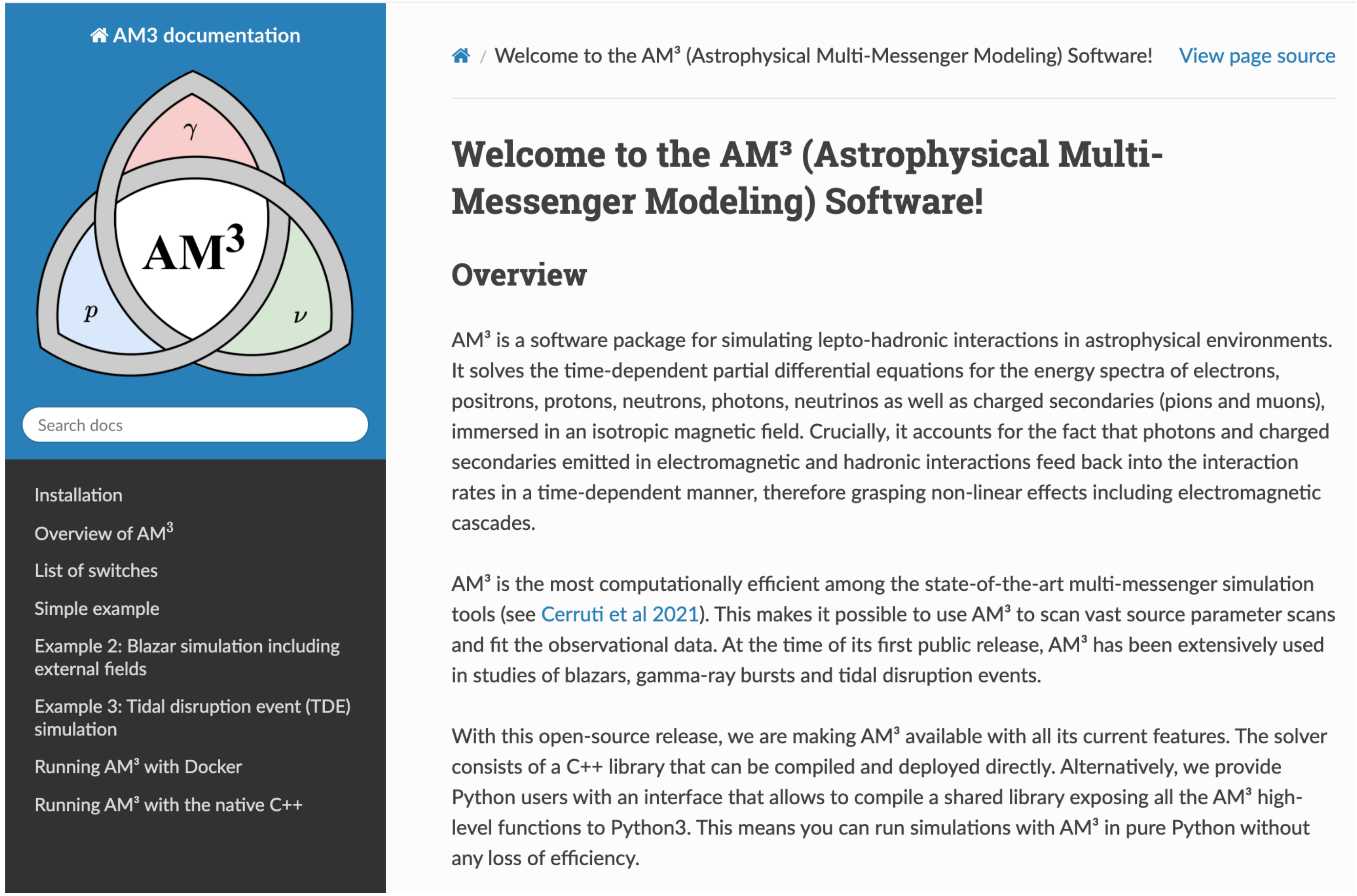
We present the AM^3 ("Astrophysical Multi-Messenger Modeling") software, which has been successfully used in the past to simulate the multi-messenger emission, including neutrinos, from active galactic nuclei, including the blazar sub-class, gamma-ray bursts, and tidal disruption events. AM^3 is a documented state-of-the-art open source software [a)] that efficiently solves the coupled integro-differential equations for the spectral and temporal evolution of the relevant particle densities (photons, electrons, positrons, protons, neutrons, pions, muons, and neutrinos). AM^3 includes all relevant non-thermal processes (synchrotron, inverse Compton scattering, photon-photon annihilation, proton-proton and proton-photon pion production, and photo-pair production). The software self-consistently calculates the full cascade of primary and secondary particles, outperforming simple test-particle approaches, and allows for non-linear feedback and predictions in the time domain. It also allows to track separately the contributions of different radiative processes to the overall photon and neutrino spectra, including the different hadronic interaction channels. With its efficient hybrid solver combining analytical and numerical techniques, AM^3 combines efficiency and accuracy at a user-adjustable level. We describe the technical details of the numerical framework and present examples of applications to various astrophysical environments.

Keywords: numerical methods — neutrino astronomy — gamma-ray astronomy — radiative processes

Now public!

https://am3.readthedocs.io/en/latest/

- paper with detailed summary of processes, solver, etc…
- documentation with examples in C++ and python


AM3 documentation
γ
AM^3
p
ν
Search docs
Installation
Overview of AM^3
List of switches
Simple example
Example 2: Blazar simulation including external fields
Example 3: Tidal disruption event (TDE) simulation
Running AM³ with Docker
Running AM³ with the native C++
/ Welcome to the AM³ (Astrophysical Multi-Messenger Modeling) Software!
View page source

Welcome to the AM³ (Astrophysical Multi-Messenger Modeling) Software!

Overview

AM³ is a software package for simulating lepto-hadronic interactions in astrophysical environments. It solves the time-dependent partial differential equations for the energy spectra of electrons, positrons, protons, neutrons, photons, neutrinos as well as charged secondaries (pions and muons), immersed in an isotropic magnetic field. Crucially, it accounts for the fact that photons and charged secondaries emitted in electromagnetic and hadronic interactions feed back into the interaction rates in a time-dependent manner, therefore grasping non-linear effects including electromagnetic cascades.

AM³ is the most computationally efficient among the state-of-the-art multi-messenger simulation tools (see Cerruti et al 2021). This makes it possible to use AM³ to scan vast source parameter scans and fit the observational data. At the time of its first public release, AM³ has been extensively used in studies of blazars, gamma-ray bursts and tidal disruption events.

With this open-source release, we are making AM³ available with all its current features. The solver consists of a C++ library that can be compiled and deployed directly. Alternatively, we provide Python users with an interface that allows to compile a shared library exposing all the AM³ high-level functions to Python3. This means you can run simulations with AM³ in pure Python without any loss of efficiency.

Now public!

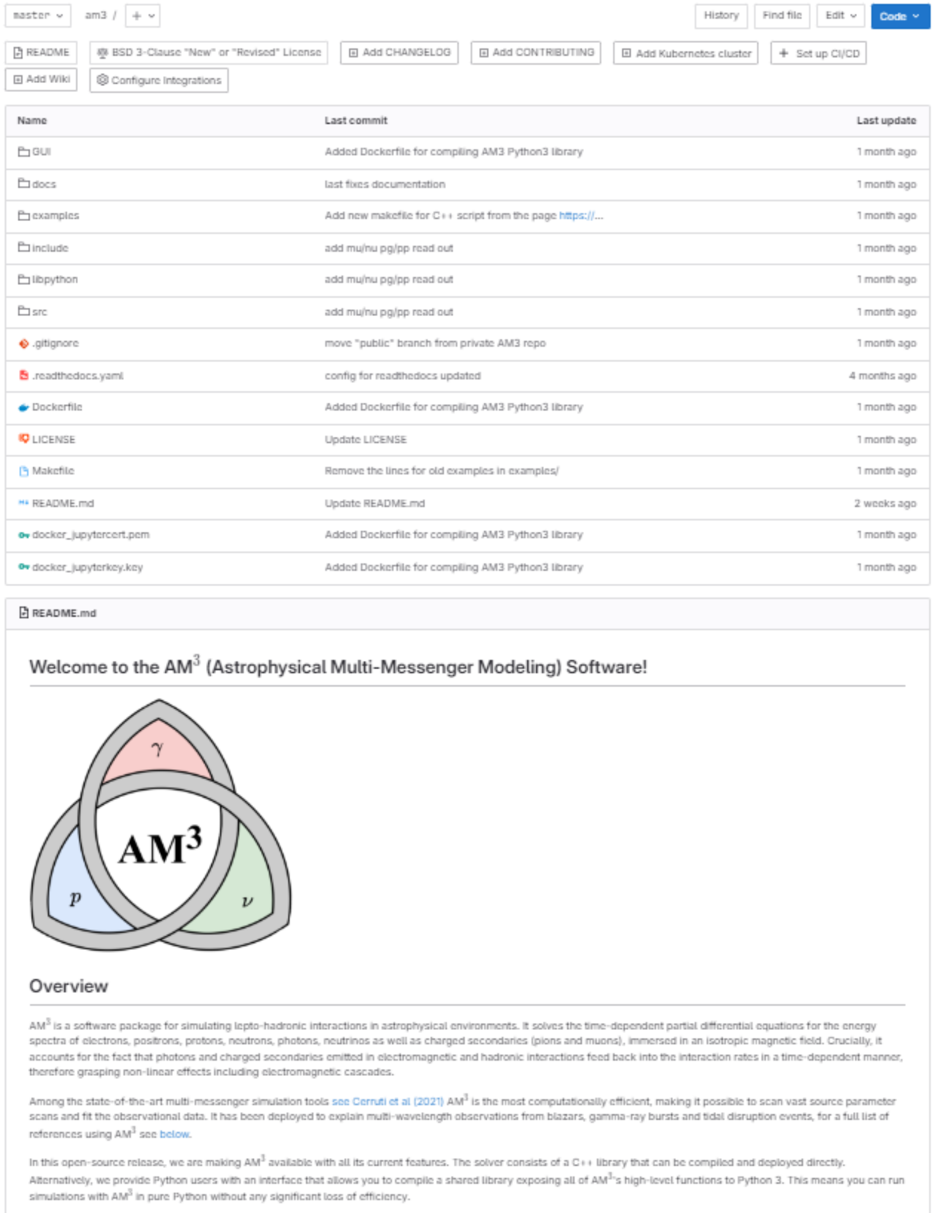
- paper with detailed summary of processes, solver, etc…
- documentation with examples in C++ and python
- e-mail address

contact-am3@desy.de

Now public!

- paper with detailed summary of processes, solver, etc…
- documentation with examples in C++ and python
- e-mail address
- source code public on GitLab and maintained by AM^3 team

https://gitlab.desy.de/am3/am3



Name	Last commit	Last update
GUI	Added Dockerfile for compiling AM3 Python3 library	1 month ago
docs	last fixes documentation	1 month ago
examples	Add new makefile for C++ script from the page https://...	1 month ago
include	add mu/nu pg/pp read out	1 month ago
libpython	add mu/nu pg/pp read out	1 month ago
src	add mu/nu pg/pp read out	1 month ago
.gitignore	move "public" branch from private AM3 repo	1 month ago
.readthedocs.yaml	config for readthedocs updated	4 months ago
Dockerfile	Added Dockerfile for compiling AM3 Python3 library	1 month ago
LICENSE	Update LICENSE	1 month ago
Makefile	Remove the lines for old examples in examples/	1 month ago
README.md	Update README.md	2 weeks ago
docker_jupytercert.pem	Added Dockerfile for compiling AM3 Python3 library	1 month ago
docker_jupyterkey.key	Added Dockerfile for compiling AM3 Python3 library	1 month ago

README.md

Welcome to the AM^3 (Astrophysical Multi-Messenger Modeling) Software!

Overview

AM^3 is a software package for simulating lepto-hadronic interactions in astrophysical environments. It solves the time-dependent partial differential equations for the energy spectra of electrons, positrons, protons, neutrons, photons, neutrinos as well as charged secondaries (pions and muons), immersed in an isotropic magnetic field. Crucially, it accounts for the fact that photons and charged secondaries emitted in electromagnetic and hadronic interactions feed back into the interaction rates in a time-dependent manner, therefore grasping non-linear effects including electromagnetic cascades.

Among the state-of-the-art multi-messenger simulation tools see Cerruti et al (2021) AM^3 is the most computationally efficient, making it possible to scan vast source parameter scans and fit the observational data. It has been deployed to explain multi-wavelength observations from blazars, gamma-ray bursts and tidal disruption events, for a full list of references using AM^3 see below.

In this open-source release, we are making AM^3 available with all its current features. The solver consists of a C++ library that can be compiled and deployed directly. Alternatively, we provide Python users with an interface that allows you to compile a shared library exposing all of AM^3's high-level functions to Python 3. This means you can run simulations with AM^3 in pure Python without any significant loss of efficiency.

Now public!

- paper with detailed summary of processes, solver, etc…

arxiv:2312.13371

- documentation with examples in C++ and python

https://am3.readthedocs.io/en/latest/

contact-am3@desy.de

- e-mail address
- source code public on GitLab and maintained by AM^3 team

https://gitlab.desy.de/am3/am3

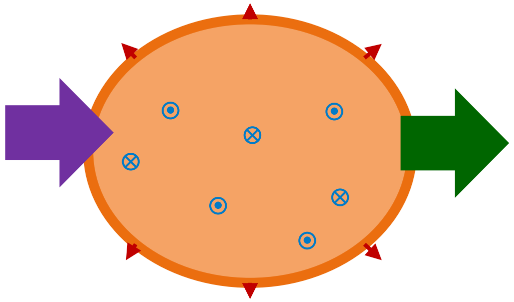
- collaborators welcome!
- C++ AND python3 (same user interface names)
- Docker

Tested

- Blazars → Gao++17,19; Rodrigues++19,21,24; Fichet de Clairfontaine++23
- TDEs → Yuan++23,24a/b
- GRB prompt emission → Rudolph++22,23a/b/c
- GRB afterglows → Klinger++24

Summary

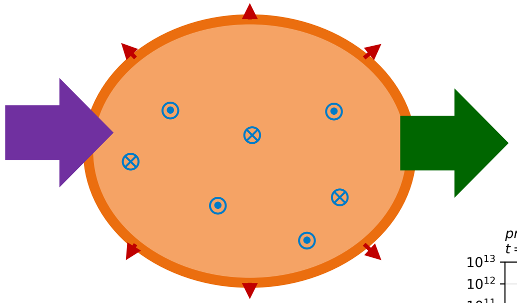
AM³ - public!

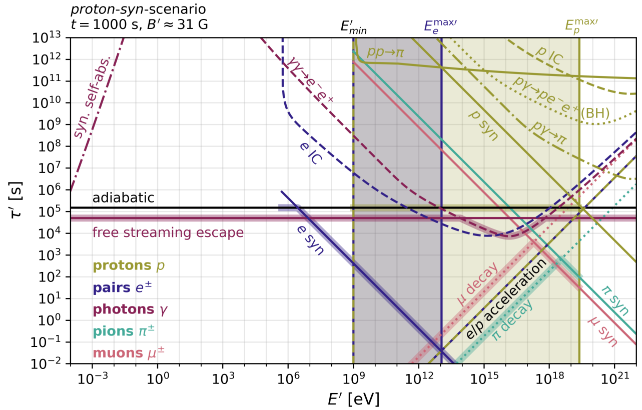


- solve transport equations - time dependent!
- for protons, electrons, photons
  + pions, muons, neutrinos

Summary

AM³ - public!



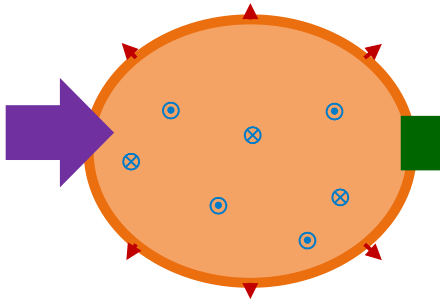


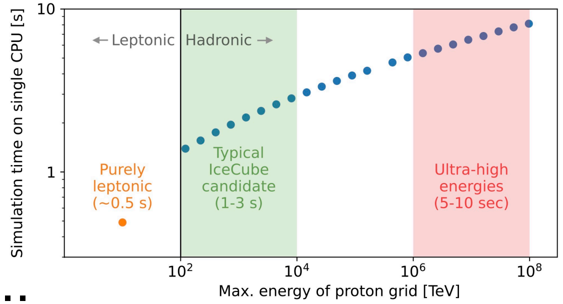
- solve transport equations - time dependent!
- for protons, electrons, photons
  + pions, muons, neutrinos
- Syn, IC, pair-prod., $p\gamma$, pp, Bethe-Heitler, decays,..

Summary

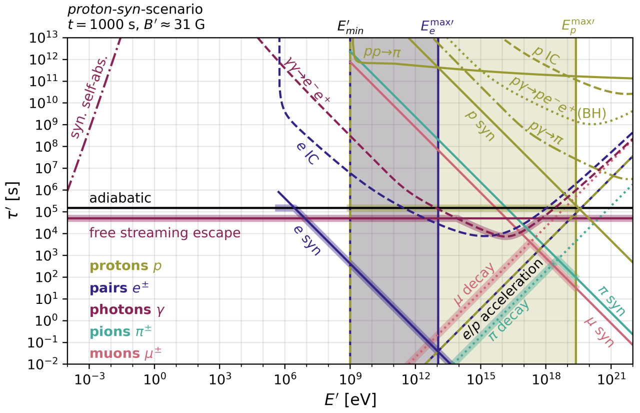
AM³ - public!

- solve transport equations - time dependent!
- for protons, electrons, photons
  + pions, muons, neutrinos
- Syn, IC, pair-prod., $p\gamma$, pp, Bethe-Heitler, decays,..
- speed optimized (steady state in ~10s)





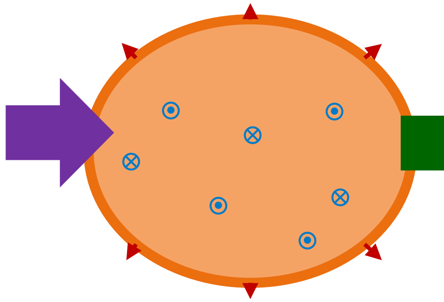
← Leptonic
Hadronic →
Purely leptonic (~0.5 s)
Typical IceCube candidate (1-3 s)
Ultra-high energies (5-10 sec)
Simulation time on single CPU [s]
10
1
10^2
10^4
10^6
10^8
Max. energy of proton grid [TeV]

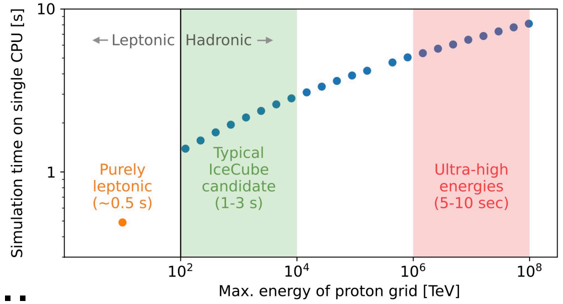


Summary

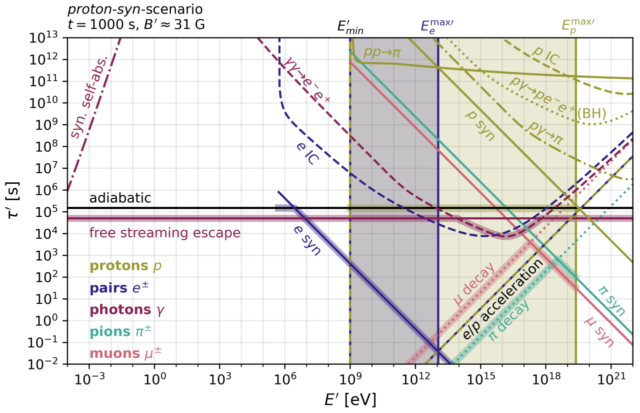
AM³ - public!

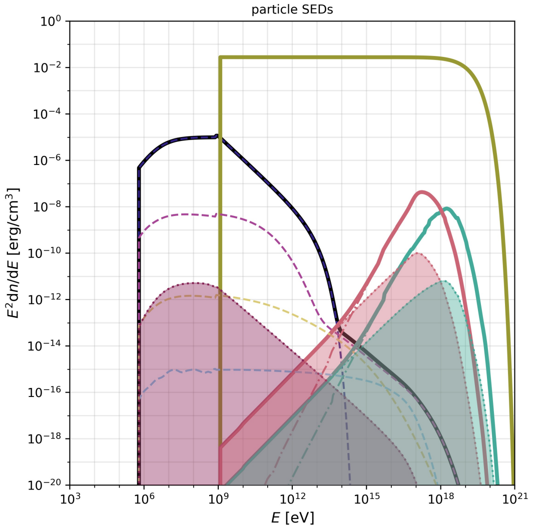
- solve transport equations - time dependent!
- for protons, electrons, photons
  + pions, muons, neutrinos
- Syn, IC, pair-prod., $p\gamma$, pp, Bethe-Heitler, decays,..
- speed optimized (steady state in ~10s)
- component tracking

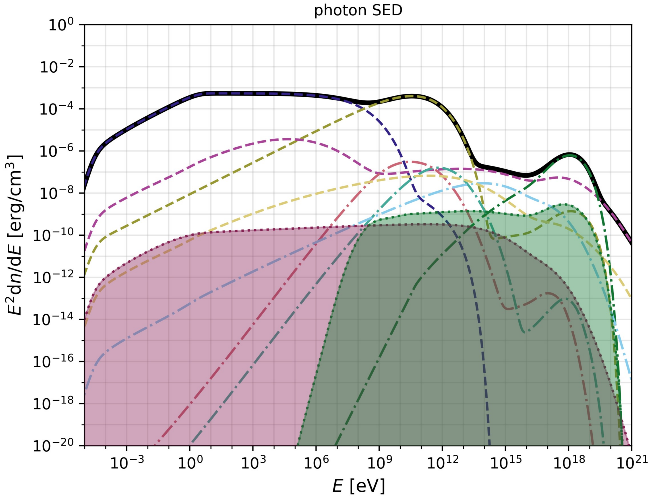




Simulation time on single CPU [s]
10
1
← Leptonic
Hadronic →
Purely leptonic (~0.5 s)
Typical IceCube candidate (1-3 s)
Ultra-high energies (5-10 sec)
10^2
10^4
10^6
10^8
Max. energy of proton grid [TeV]





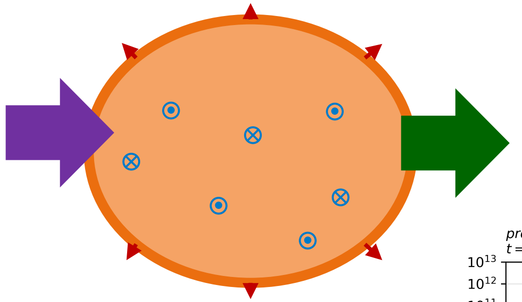


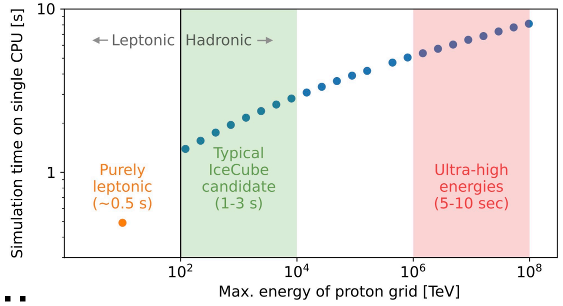
Summary

AM³ - public!

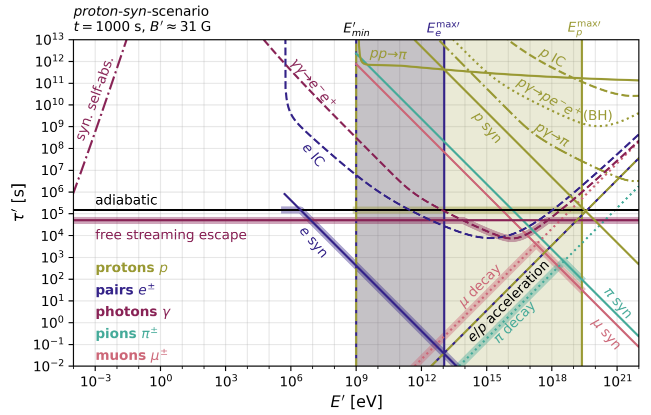
- solve transport equations - time dependent!
- for protons, electrons, photons
  + pions, muons, neutrinos
- Syn, IC, pair-prod., $p\gamma$, pp, Bethe-Heitler, decays,..
- speed optimized (steady state in ~10s)
- component tracking
- written in C++, interface to python
- used already for blazars (initially Gao++ 2017), GRBs, TDEs
- public including documentation!

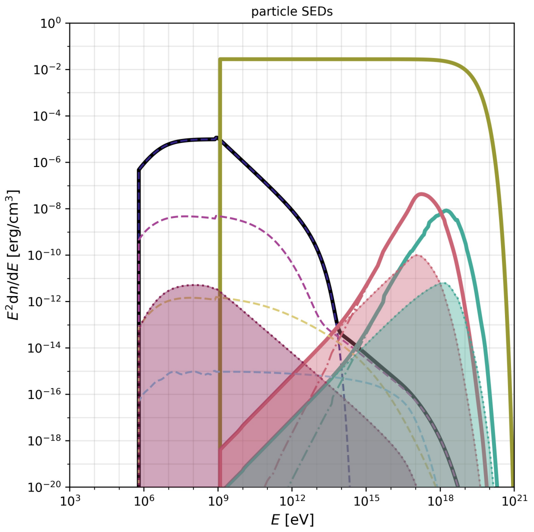
https://gitlab.desy.de/am3/am3

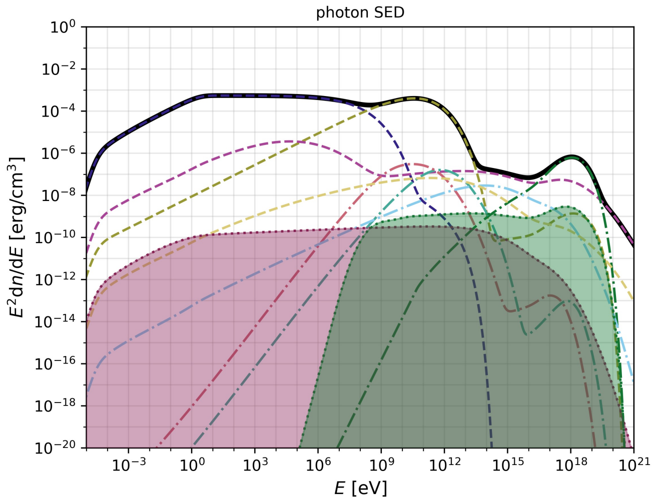




Simulation time on single CPU [s]
10
1
← Leptonic
Hadronic →
Purely leptonic (~0.5 s)
Typical IceCube candidate (1-3 s)
Ultra-high energies (5-10 sec)
10^2
10^4
10^6
10^8
Max. energy of proton grid [TeV]







Backup

Estimating the time scales / source terms

process: $a \to b$ (e.g. synchrotron)

$$Q_{a\to b}(E_b, t) = \overbrace{\int \mathrm{d}\ln E_a n_a(E_a)}^{\text{convolution}} \overbrace{R_{a\to b}(E_a, E_b)}^{\text{radiation kernel}} \qquad \alpha_{a\to b}(E_a, t) = \overbrace{\int \mathrm{d}\ln E_b n_b(E_b)}^{\text{convolution}} \overbrace{R_{a\to b}(E_a, E_b)}^{\text{radiation kernel}}$$

process: $a + b \to c + d$ (e.g. inverse Compton)

$$Q_{a,b\to c}(E_c, t) = \overbrace{\int \mathrm{d}\ln E_a n_a(E_a)}^{\text{convolution}} \overbrace{\int \mathrm{d}\ln E_b n_b(E_b)}^{\text{convolution}} R_{a,b\to c}(E_a, E_b, E_c)$$

$$\underbrace{R_{a,b\to c}(E_a, E_b, E_c)}_{\text{radiation kernel}} = \frac{c}{2} \underbrace{\int \mathrm{d}\mu\,(1-\mu)}_{\text{angle average}} \underbrace{\frac{\mathrm{d}\sigma_{a,b\to c}}{\mathrm{d}E_c \mathrm{d}\mu}(E_a, E_b, E_c, \mu)}_{\text{cross section}}$$

$$\alpha_{a,b\to c}(E_a, t) = \overbrace{\int \mathrm{d}\ln E_c n_c(E_c)}^{\text{convolution}} \overbrace{\int \mathrm{d}\ln E_b n_b(E_b)}^{\text{convolution}} R_{a,b\to c}(E_a, E_b, E_c)$$