1 RT^2 Input Format

The input system of RT^2 was developed with reference to the input formats of widely used Monte Carlo codes such as FLUKA and MCNP. Compared to these Fortran-based codes, the input format of RT^2 is relatively flexible.

An RT2 input file consists of multiple cards. In general, the order of the cards is not strictly constrained, but dependencies between them may require certain cards to appear in a specific order.

If the first non-whitespace character of a line is '#', the line is treated as a comment. Lines containing only whitespace are also ignored.

Each card follows the format described below.

```
CARD_KEY  --arg1   value1   --arg2   value2  …
```

CARD_KEY specifies the type of the card. arg1, arg2, … represent the names of the arguments associated with the card, and values for each variable can be assigned using the following strings.

Each variable must be separated by whitespace. The number of spaces and the use of tab characters are flexible and may be mixed.

The order of the arguments is not important. Some arguments may require multiple input values, in which case the values should be entered consecutively, separated by spaces.

An example is shown below.

```
CARD_KEY  --arg1    1.5   0.0   0.0  …
```

In this example, a 3-element vector {1.5, 0.0, 0.0} is assigned to the variable arg1.

Each argument associated with a card must receive a fixed number of values. If the number of provided values does not match the required length, an error may occur.

If a card becomes excessively long, the line can be continued by placing a backslash (\) at the end.

An example is shown below.

```
CARD_KEY  --arg1  value1   --arg2  value2   \
          --arg3  value3   --arg4  value4   \
          --arg5  value5   --arg6  value6
```

2 List of RT2QMD input cards

2.1 CUDA_SETTINGS

The CUDA_SETTINGS card is used to configure GPU-related settings for the overall execution of the Monte Carlo code.

The available arguments are as follows:

- gpu [int]: The ID of the GPU on which the code run. GPU list can be found in command 'nvidia-smi'. (Default=0)

- block_dim [int]: The number of threads in a single block. The total number of threads is product of the block_dim, block_per_sm, and the number of stream multiprocessor (SM). Here, the number of SM is determined by hardware specification. (Default=128)
- block_per_sm [int]: The number of blocks per stream multiprocessor. If the metric 'Volatile GPU-Util' in the 'nvidia-smi' is not saturated, block_dim should be increased to achieve peak performance. (Default=48)
- buffer_ratio [float]: The ratio of memory share of particle and interaction buffer per total GPU memory capacity. If the error caused by the buffer size is raised, this parameter should be increased. (Default=0.6)
- block_decay_rate [float]: The decay rate of the block of launched kernel at the residual stage. This parameter can be tuned heuristically to achieve peak performance. Generally, this parameter should have a value close to 1 when activating a computationally intensive interaction (e.g. Quantum Molecular Dynamics). (Default=0.2)
- block_decay_limit [int]: The lower bound on the number of the block in the launched kernel. When the decayed number of blocks become smaller than this limit, all remained phase-space of each buffer will be discarded. Peak performance can be achieved by increasing this parameter, as small kernel launch sizes degrade performance. However, accuracy may decrease because remaining particles are removed. (Default=1)
- seed [int] : Initial seed of the CURAND XORWOW random number generator. (Default=42)

2.2 QMD_SETTINGS

This card configures the QMD model.

- nn_scattering [str]: Selects the model used for NN elastic scattering. The available options are:

  geant4: Uses a tabulated NN scattering table (Default)

  incl: Uses an equation-based scattering algorithm from the Liège intranuclear cascade model.
- measure_time [bool]: Enables output of execution cycles for specific kernels. To use this option, DMCRT2_ACTIVATE_TIMER=1 must be set during the CMake configuration step. (Default=false)
- timer_size [int]: The number of kernels to be timed. If this count is exceeded, timing is skipped. (Default=20)
- dump_action [bool]: Dumps the mean-field matrix and the phase-space of member nucleons at each time step. Only the first block generates the dump. For performance reasons, this option is only available in debug configuration and has no effect in the release mode. (Default=false)
- dump_size [int]: Number of dump entries to be saved. (Default=1000)

2.3 MODLUE_TEST

This card configures the event generator.

- event [str]: Selects the mode of the event generator. The available options are:

  QMD: Simulate the Quantum Molecular Dynamics (Default)

  deex: Simulate the de-excitation chain.

- nps [int]: Number of events to compute per iteration. (Default=1e6)
- iter [int]: Number of times the event generator will be executed. (Default=1)
- full_mode [bool]: If enabled, the calculation proceeds until all secondaries reach their ground state. If disabled, only the first applicable model is executed. For example, if this parameter is set to false in QMD mode, the de-excitation step is skipped and only the phase-space of excited secondaries is returned. (Default=false)
- write_ps [bool]: If enabled, the phase-space data of all generated secondaries is saved as binary files. Repeated disk I/O operations may significantly impact performance. (Default=false)
- hid [bool]: Tracks the event ID that produced each secondary in the phase-space output. If disabled, the origin event of each secondary is not recorded. This may be useful for checking energy and momentum conservation, but it can slightly affect performance. (Default=false)

2.4 EVENT

Defines the properties of the event to be simulated. When multiple EVENT cards are provided, events are randomly sampled according to the specified weight ratios. At least one EVENT card is required.

- position [double3]: Spatial position where the event occurs [cm]. (Default={0.0,0.0,0.0})
- direction [double3]: Momentum unit vector of the projectile (QMD). Momentum of the excited nucleus (deex), in unit of [MeV/c]. (Default={0.0,0.0,1.0})
- energy [double]: Kinetic energy of the projectile [MeV/u] (QMD). Excitation energy of the nucleus [MeV] (deex).
- weight [double]: Sampling weight of the event. Automatically normalized across all events. (Default=1.0)
- zap [int2]: ZA number of the projectile nucleus.
- zat [int2]: ZA number of the target nucleus. Used only in QMD mode.

2.5 YIELD

Score energy-angle double-differential yield of the secondary particle.

- name [str]: The name of tally. This variable must be unique.
- part [str list]: Particle namelist of tally filter. Possible options are followed as below:

  all: Scoring all particles

  photon, g: Photon

  neutron, n: Neutron

  ion, genion: Generic ion (include proton, deuteron …)
- escale [str]: Energy binning type of this tally. Possible options are followed as below

  linear: The energy bin is linearly uniform

  log: The energy bin is log uniform
- erange [double2]: Lower and upper bound of energy bin [MeV/u]. Upper bound should be larger than lower bound.

- ne [int]: The number of energy bin. This value must be positive.
- dtheta [double2]: polar angle of the estimator [degree]. Initial projectile direction is assumed to be z-axis aligned, and secondary particles are regard as azimuthal symmetry. Upper bound should be larger than lower bound.
- ascii [bool]: Write tally data as ASCII if true, as binary elsewhere. (Default=false)
- za [int list]: A list of ZA filter of this estimator. This parameter is used when the 'part' includes the generic ion. The format of this parameter is ZZZAAA. If A = 0, all particles matching the specified Z number will be scored. For example, '--za 1000 2000' scores all hydrogen and helium isotopes.