# Progress of BdNMC work

Shaghayegh Atashi
June 15, 2017

## Outline

- Some info on BdNMC production and scattering channels
- TTree that stores all the information in events.dat
- A bit more on elastic NC-like nucleon scattering

# Production channel: pi-minus_capture

- In the description of the BdNMC 3.1.5 (newer version), Patrick says the piminus capture signal channel is added … "Piminus scattering simulates the isotropic emission of dark photons from the pi-minus absorption process P+pi minus -> N* -> N + \gamma. This production channel can be invoked with production_channel piminus_capture."
  - I'm assuming N= proton or neutron as usual
- According to the comments in the sample BdNMC parameter card, **piminus_capture is a baryonic production channel.**
  - **Problem: BdNMC gives a negative number of V's, then produces a segmentation fault with this production channel**
  - **But: we can ignore this production channel for now (dark photons are emitted isotropically and SK is far)**

# Production channel: parton_production

- Corresponds to the dark matter production process $p+N \rightarrow V^* \rightarrow \chi/\chi^\dagger$
- I want to see what these events look like in events.dat
- I get a seg fault when I run BdNMC with the parton_production mode:

“Parameter read successfully

Setting up *distribution default* for channel parton_production

./bin/BDNMC: line 26:  6912 Segmentation fault      ./build/main $RUN_DIRECTORY/$arg”

  - BdNMC has trouble with the default distribution for parton_production. I'm running this production channel without specifying a production distribution (specifying a production channel is “optional”, but it seems like  I need a production distribution)

    Paper talks about the production channel distributions:

    - “parton_V” for parton_production production channel
    - But: “this requires externally generated data for V-production at the parton level”
    - I'll have to find this data

# More production channels

- Appendix says these are also possible production channels: omega (omega in omega_decay), rho (in rho_decay) but when I run it with omega_decay, I get “No DM production expected”

  -> (maybe there is DM production for some model parameters, but I don't know what)

- I get a seg fault when I run it with rho_decay

=> these production channels are probably not yet implemented in BdNMC

# Signal channel: inelastic NC $\pi^0$-like nucleon scattering

(for all production channels) it's possible that the produced $\chi$ undergoes inelastic NC $\pi^0$ -like nucleon scattering:

- a situation "where there is sufficient momentum transfer to produce a neutral pion which subsequently decays producing a two-photon signature (also one of the main background for nu_mu → nu_e appearance)

- "Incoherent NC$\pi^0$ ; pion emerges via the production of a $\Delta$ (1232) resonance in the following process

$\chi + N \rightarrow \chi + \Delta$

$\Delta \rightarrow N + \pi^0$

An example event:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| event 1 | | | | | | | | |
| pion | -0.0836232 | 0.00222933 | 4.75508 | 4.75774 | | | | |
| V | -0.0424542 | 0.00297251 | 2.61895 | 2.6212 | | | | |
| DM | -0.0152323 | -0.00434501 | 2.41499 | 2.41506 | | | | |
| Recoil_DM | 0.151803 | -0.0496552 | 2.05259 | 2.05882 | | | | |
| Delta | -0.167035 | 0.0453101 | 0.362395 | 1.2958 | -3.13666 | -0.894731 | 497.297 | 1.65885e-06 |
| Neutron | -0.144091 | 0.0478925 | 0.048097 | 0.95297 | | | | |
| pi0 | -0.0229438 | -0.00258233 | 0.314298 | 0.342834 | | | | |

# Inelastic NC $\pi^0$-like nucleon scattering cont.

**Some things to think about:**

- Somewhere else in the paper [1] , it says that the pi0 → V → pair of DM particles decay process happens on such a short scale that the pi0 and V don't travel far, so we can assume this process happens in the target and the DM particles propagate from the target to the detector.

- It seems like it's possible that the pi0 from the inelastic scattering inside the detector decays (in flight) to a V which decays to 2 DM particles all inside the detector, so the DM particles scatter inside the detector. Then the DM particles can undergo scattering in the detector.

  - One scenario: the DM particles can undergo inelastic NC pi0 -like scattering again and produce another pi0.
  - The process can repeat and produce a cascade of DM particles that scatter inside the detector?

→ This is too unlikely to happen; pi0 → 2 gamma's is much more probable

1- "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniooNE, T2K, and SHiP" by Patrick de Niverville et al.

# Leptophobic model of DM

- Discussion with Hiro: I'll focus on the dark photon model for now; can consider the baryonic vector mediator model separately later

# Putting BdNMC's output information into a TTree:

- Reminder:
  BdNMC outputs a data file containing info about the particles involved in the interactions. For example, some events from a data file outputted by BdNMC is shown here. The format is:

  particle name px py pz E x y z t

```
Run 1495052956
event 1
eta        0.259824     -0.198728     3.08678    3.15205
V          0.267948     -0.0747792    2.75542    2.79816
DM         -0.00796544  0.00861999    1.8036     1.80364
proton     -0.342382    -0.192742     0.132485   1.02581    -2.1939    2.37418     496.759    1.65705e-06

endevent 1

event 2
eta        -0.512015    -0.241737     2.33848    2.46764
V          -0.447128    -0.203878     2.29495    2.38081
DM         0.00205929   -0.0015099    0.924071   0.924088
proton     -0.297628    -0.242418     0.186791   1.03082    1.10585    -0.810828   496.231    1.65528e-06

endevent 2
```

- x,y,z, and t are only shown for the signal particle (particle that the DM scatters off)

# Ttree that stores all the information in events.dat

- Branches that are arrays of variable length
- The branches:

  event_num: event number

  n: number of particles in each interaction; varies for each event

  eg **n=3 for event 1**

  **n=4 for event 2** in the events.dat file below

```
event 1
V         -0.447128    -0.203878    2.29495    2.38081
DM        0.00205929   -0.0015099   0.924071   0.924088
proton    -0.297628    -0.242418    0.186791   1.03082    1.10585    -0.810828   496.231   1.65528e-06

endevent 1

event 2
eta       0.259824     -0.198728    3.08678    3.15205
V         0.267948     -0.0747792   2.75542    2.79816
DM        -0.00796544  0.00861999   1.8036     1.80364
proton    -0.342382    -0.192742    0.132485   1.02581    -2.1939    2.37418     496.759   1.65705e-06
```

# Ttree that stores all the information in events.dat

- The other branches (they are all arrays of size n):
- Nth part: an integer that stores the order of the particle in the interaction (eg 1 for the first particle in the interaction, 2 for the second, etc)

  → useful later when I want to know which particles participated in each interaction for a given event
- The particle type (a Char_t)
- An integer corresponding to the particle type (using PDG numbering convention)
  - Useful when putting the scattered particles into skdetsim
- px: x momentum (GeV)
- py: y momentum (GeV)
- Pz: z momentum (GeV)
- E: energy (GeV)
- x: x position of interaction (m)
- y: y position of interaction (m)
- z: z position of interaction (m)
- time: time of interaction (since production of dark matter particles, in seconds)

# TTree cont.

- Want to fill position and time branches when the particle is a signal particle **(eg the neutrons below)**
- Run into issued if I try to fill the branches when position/time is not give for the particle (eg for eta and DM below)

```
event 4
eta       -0.293065    0.34329      0.469601    0.851124
V         -0.309001    0.180154     0.415015    0.678362
DM        0.000520242  0.000693986  0.303537    0.30358
neutron   0.179911     -0.149193    0.206807    0.988812   0.854665    1.14009    498.657   1.66357e-06
DM        0.0520242    0.000893986  0.4303537   0.230358
neutron   0.479911     -0.249193    0.4206807   0.198881   0.3854665   1.514009   198.657   2.66357e-06

endevent 4
```

→ I have separate branches that only get filled for the signal particles

# TTree cont.

- Read the file once to determine # particles in each event
  - Also determine #scattering particles (by looking at the size of the line) and store them in a vector
- The branches:

n :#particles in each event

- Nth part
- particle type (Char_t)
- Int corresponding to particle type
- px (GeV)
- py (GeV)
- pz (GeV)
- E (GeV)

These are arrays of size n

n_scatt: # scattering particles in each event

- Nth part (for scattering particles)
- particle type (Char_t) (for scattering particles)
- Int corresponding to particle type (for scattering particles)
- px (for scattering particles) (GeV)
- py (for scattering particles) (GeV)
- pz (for scattering particles) (GeV)
- E (for scattering particles) (GeV)
- X (in m, for scattering particles)
- Y (in m, for scattering particles)
- Z (in m, for scattering particles)
- Time (in s, for scattering particles)

These are arrays of size n_scatt

These branches only get filled for signal particles (using the size of the line)

# TTree cont.

## Integer-based encoding of particles

- I've created 2 vectors (elements are in correspondence)

  - one vector stores particle names

  - the other stores the corresponding integer

- For each particle name (in events.dat):
  - loop through all entries in the names vector until the names match (let's say at index i), the corresponding integer is = integers_vector[i]

# pion_inelastic and pion_inelastic_charged signal channels

- Reminder:

  pion_inelastic (NCpi0-like inelastic scattering):

  $\chi + N \rightarrow \chi + \Delta$

  $\Delta \rightarrow N + \pi^0$

- Eg:

| event 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| pion | -0.0836232 | 0.00222933 | 4.75508 | 4.75774 | | | | |
| V | -0.0424542 | 0.00297251 | 2.61895 | 2.6212 | | | | |
| DM | -0.0152323 | -0.00434501 | 2.41499 | 2.41506 | | | | |
| Recoil DM | 0.151803 | -0.0496552 | 2.05259 | 2.05882 | | | | |
| Delta | -0.167035 | 0.0453101 | 0.362395 | 1.2958 | -3.13666 | -0.894731 | 497.297 | 1.65885e-06 |
| Neutron | -0.144091 | 0.0478925 | 0.048097 | 0.95297 | | | | |
| pi0 | -0.0229438 | -0.00258233 | 0.314298 | 0.342834 | | | | |

| event 12 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| pion | -0.0296693 | -0.0510476 | 1.8827 | 1.88846 | | | | |
| V | -0.0490881 | -0.030156 | 1.64648 | 1.65052 | | | | |
| DM | -0.00990177 | -0.00416781 | 1.3567 | 1.35678 | | | | |
| Recoil DM | 0.172118 | 0.273436 | 0.874492 | 0.932324 | | | | |
| Delta | -0.18202 | -0.277604 | 0.482204 | 1.36402 | -3.6425 | -1.53319 | 499.078 | 1.66484e-06 |
| Proton | -0.0131495 | -0.0815221 | 0.21368 | 0.965832 | | | | |
| pi- | -0.16887 | -0.196082 | 0.268524 | 0.398184 | | | | |

- The nucleon and pion after delta are its decay products
- I determine the charge of delta using its decay products and store it in my branches
- Same thing with the Pion_Inelastic_Charged signal channel (next slide)

# pion_inelastic and pion_inelastic_charged signal channels

- pion_inelastic signal ch: also includes Δ decay to charged pion states
- Eg:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| event 2 | | | | | | | | |
| pion | -0.0485683 | -0.131048 | 3.98058 | 3.98532 | | | | |
| V | -0.0142828 | -0.0707355 | 2.78356 | 2.78629 | | | | |
| DM | 0.00722674 | 0.000436931 | 1.7374 | 1.73744 | | | | |
| Recoil DM | -0.193723 | 0.114637 | 1.3465 | 1.36522 | | | | |
| Delta | 0.20095 | -0.1142 | 0.3909 | 1.31303 | 2.07526 | 0.125471 | 498.919 | 1.66426e-0 |
| Proton | 0.364715 | -0.187031 | 0.27344 | 1.05977 | | | | |
| pi0 | -0.163765 | 0.0728302 | 0.11746 | 0.253256 | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| event 6 | | | | | | | | |
| pion | -0.150926 | 0.0187367 | 3.5458 | 3.55163 | | | | |
| V | -0.136918 | -0.0105639 | 3.09401 | 3.09867 | | | | |
| DM | -0.0050204 | 0.00114232 | 1.18229 | 1.18235 | | | | |
| Recoil DM | -0.162419 | 0.0577219 | 0.806967 | 0.825232 | | | | |
| Delta | 0.157399 | -0.0565796 | 0.375328 | 1.29872 | -2.11355 | 0.480909 | 497.737 | 1.66035e-0 |
| Neutron | 0.142721 | -0.122292 | 0.516997 | 1.08876 | | | | |
| pi+ | 0.0146779 | 0.0657127 | -0.14167 | 0.209961 | | | | |

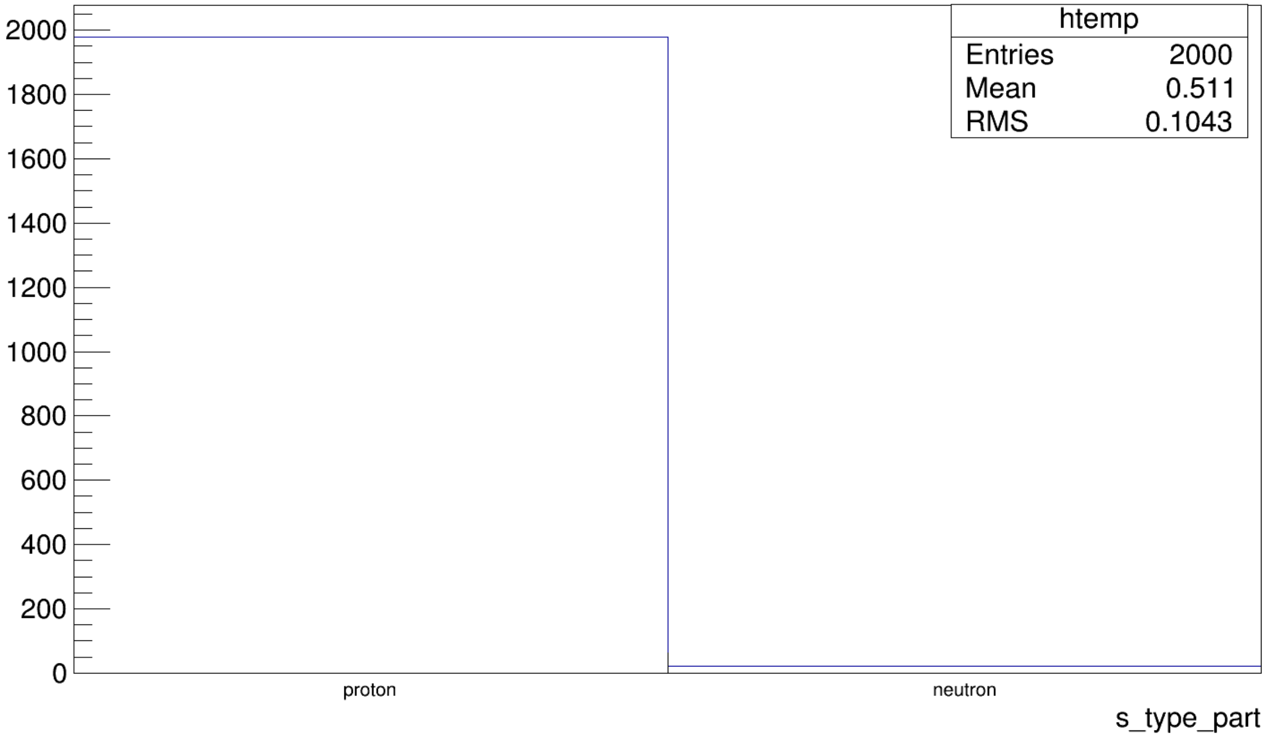
→ Determine charge of delta from its decay products

# More on elastic NC- like nucleon scattering

- Eg: running BdNMC with a MiniBooNE- like experiment with

**epsilon = 1e-3** **dark_matter_mass= 0.01 GeV** **dark_photon_mass = 0.1 GeV** **alpha_D = 0.1**

s_type_part (Type of scattering particle)




**→ a lot more proton scattering**

# More on elastic NC- like nucleon scattering

- Elastic NC-like nucleon scattering:

  The paper says for incoherent scattering, the leading term of the cross section is

$$\frac{d\sigma_{\chi N}}{dE_\chi} = 4\pi k_{V,B}^{(1)} Q_N^2 G_D(Q^2) \frac{2m_N E E_\chi - m_\chi^2 (E - E_\chi)}{(E^2 - m_\chi^2)(m_V^2 + Q^2)^2} + \cdots \quad (13)$$

where again $Q_N$ is the nucleon electric charge (or unity in the baryonic case), $E_\chi$ the energy of the recoiling DM particle, $Q^2 = 2m_N(E - E_\chi)$ is the momentum transfer, and $G_D(Q^2)$ is the Sachs form-factor, $G_D(Q^2) = 1/(1 + Q^2/M^2)^2$ with $M = 0.843\,\text{GeV}$. Further dipole form factor terms, which are generally subleading (for protons), are suppressed to simplify the presentation, although they are included in the final results (see [2, 43] for full details). The resulting nuclear scattering cross section will be discussed later in Section 5.

$$k_{V,B}^{(n)} = \begin{cases} \epsilon^2 \alpha (\alpha')^n & \text{for } U(1)' \\ \alpha_B^{n+1} & \text{for } U(1)_B \end{cases}$$

g' is the U(1)' gauge coupling

$\alpha' \equiv g'^2/(4\pi)$

  - > leading term couples to electric charge so more proton scatterings than neutron scatterings
  - Leading term is dependent on mass of chi and dark photon.

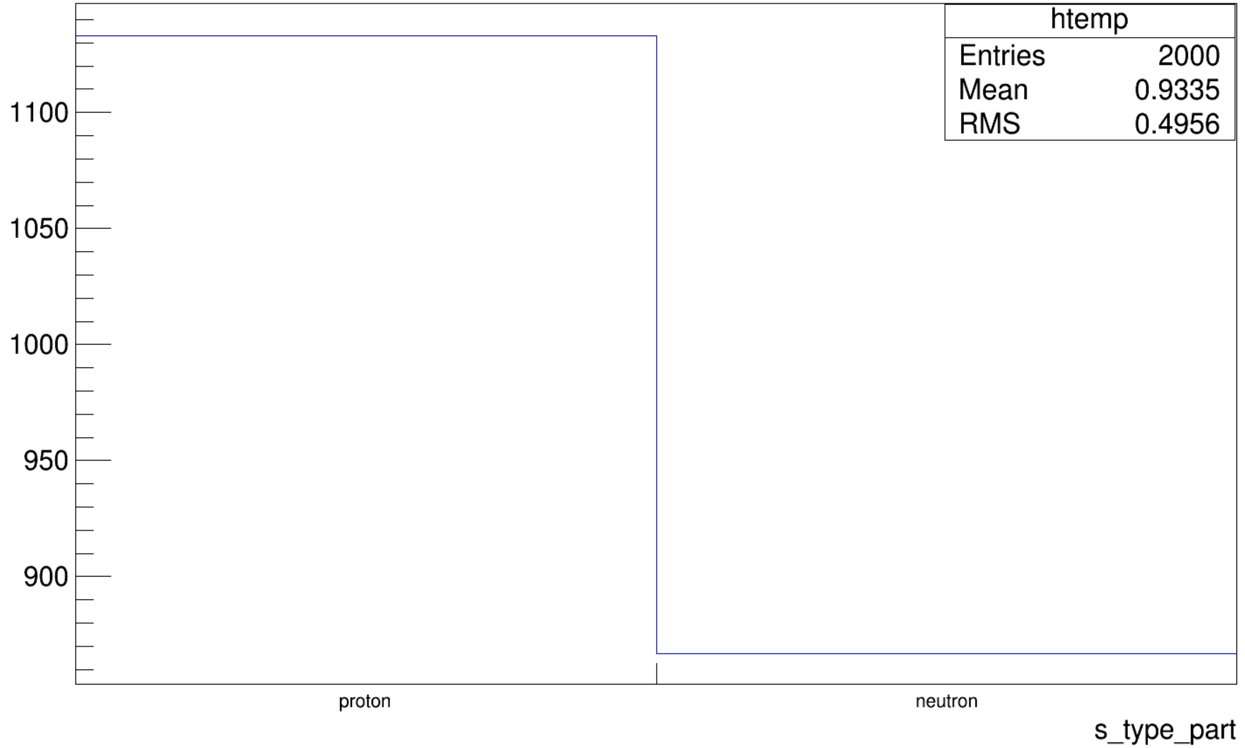
# Elastic NC-like nucleon scattering cont.

$$\frac{d\sigma_{\chi N}}{dE_\chi} = 4\pi k_{V,B}^{(1)} Q_N^2 G_D(Q^2) \frac{2m_N E E_\chi - m_\chi^2(E-E_\chi)}{(E^2-m_\chi^2)(m_V^2+Q^2)^2} + \cdots \quad (13)$$

where again $Q_N$ is the nucleon electric charge (or unity in the baryonic case), $E_\chi$ the energy of the recoiling DM particle, $Q^2 = 2m_N(E - E_\chi)$ is the momentum transfer, and $G_D(Q^2)$ is the Sachs form-factor, $G_D(Q^2) = 1/(1 + Q^2/M^2)^2$ with $M = 0.843\,\text{GeV}$. Further dipole form factor terms, which are generally subleading (for protons), are suppressed to simplify the presentation, although they are included in the final results (see [2, 43] for full details). The resulting nuclear scattering cross section will be discussed later in Section 5.

- Ran the same parameter card except with NCE_nucleon_baryonic: thought the asymmetry would disappear …

s_type_part (Type of scattering particle)




→ **Will look at other terms in the scattering cross section to get a better idea**

# Next steps

- Run my code (event.dat → TTree) with different BdNMC parameters and study the histograms
- make cuts on the data that enters the histograms by calling MakeClass() on the tree (eg make seperate histograms for scattered protons vs neutrons etc)

# Backup

# pion_inelastic and pion_inelastic_charged signal channels

- Reminder:

  pion_inelastic (NCpi0-like inelastic scattering):

  $\chi + N \rightarrow \chi + \Delta$

  $\Delta \rightarrow N + \pi^0$

- pion_inelastic signal ch: also includes Δ decay to charged pion states
  - occurs ~ 1/3 of the time, increasing the event rate over Pion_Inelastic by 50%.

- Charged pion decays:

$$\pi^+ \rightarrow \mu^+ + \nu_\mu, \quad \pi^- \rightarrow \mu^- + \bar{\nu}_\mu$$

Branching ratio: 0.999877

$$\pi^+ \rightarrow e^+ + \nu_e, \quad \pi^- \rightarrow e^- + \bar{\nu}_e$$

Branching ratio: 0.000123

$$\pi^- \rightarrow \pi^0 + e^- + \bar{\nu}_e, \quad \pi^+ \rightarrow \pi^0 + e^+ + \nu_e$$

Branching ratio: ~ $10^{-8}$

# Next steps cont.

- What is this?

```
#This invokes the bremsstrahlung production channel. This works, but may be
#unreliable around the rho resonance. The zmin/zmax values seem reasonable
#for MiniBooNE energies. ptmax could be as large as the proton mass, but
#probably would not change signal much.
production_channel V_decay
production_distribution proton_brem
ptmax 0.2
zmin 0.3
zmax 0.7
```

# Title title title title title title title title title

- text