UNIVERSITY OF TORONTO

T2K

Weekly Update

Shaghayegh Atashi

July 10, 2017

Outline

- Some info on the partonic production channel
- How BdNMC calculates # of signal events
- Reproducing some plots from paper

Update on parton_production channel

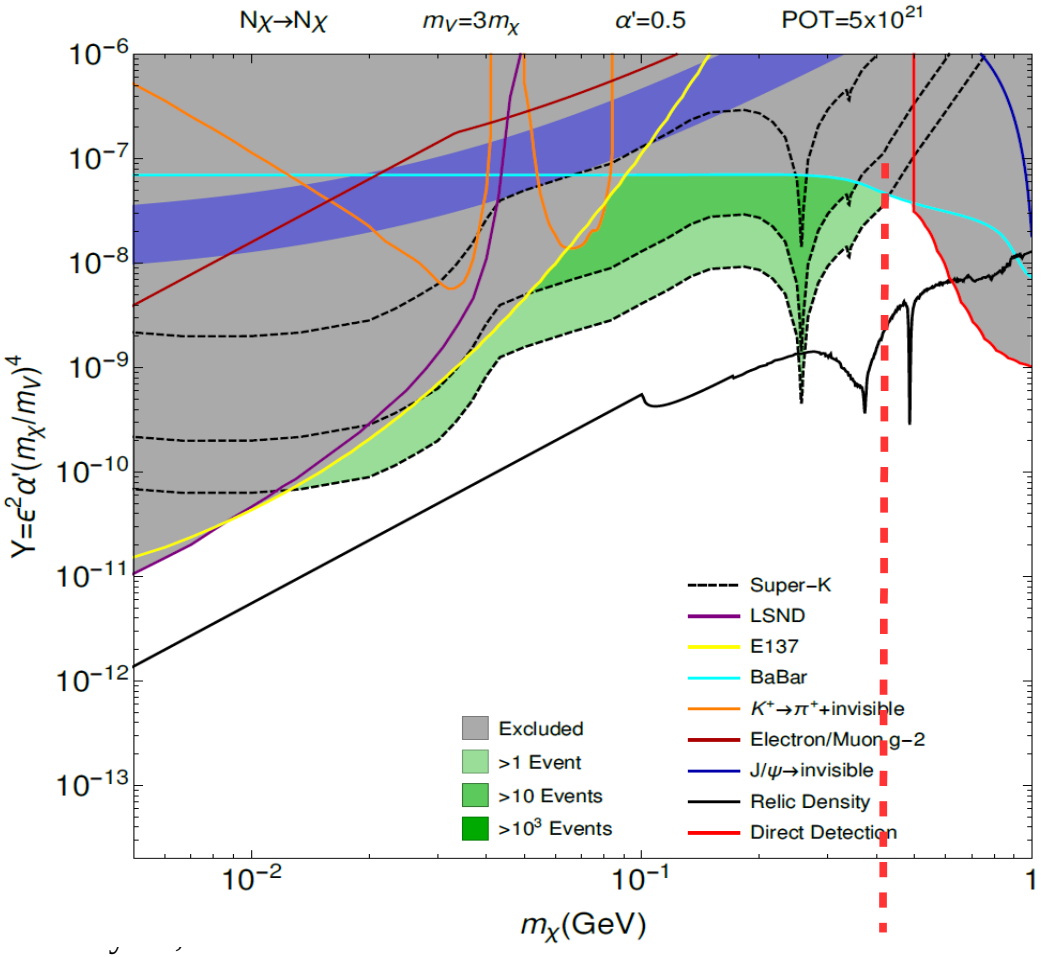
- Reminder: corresponds to the dark matter production process $p+N \rightarrow V^* \rightarrow \chi/\chi^\dagger$
- Need 2 externally generated cvs files
  - Patrick can make these

Update on parton_production channel cont.

Question: Is this production channel relevant for SK?

- Partonic production relevant for $m_V >$ 1GeV [1]

SK sensitivity plot – figure 10 of paper [1]



(with $m_V=3m_x$, alpha'=0.5, POT = $5x10^{21}$)
sensitive up to $m_X \sim 0.7$ GeV

=> $m_V < 2.1$ GeV

- Couldn't find dependence of partonic production on beam energy ...

[1] "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniBooNE, T2K and SHiP" by deNiverville et al.

How BdNMC calculates # of signal events

- I will show an overview of the simulation loop and how the # of signal events is calculated
- Before the simulation loop proper begins, a burn-in run is conducted for each production channel to estimate pmax, the maximum scattering probability encountered.
  - Very similar to the simulation loop, but the end state interaction results are not generated, and the variable nburn is incremented each time. The burn-in run is conducted until burntrials is equal to BURNMAX.

An overview of the simulation loop:

- samplesize is a parameter chosen by the user
- The code conducts trials until samplesize scattering events occur inside the detector
- ninteractions records the # of scattering events inside the detector, for each production channel

More detailed overview of the simulation loop:

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.

Initialize everything

2. While `nevent` $\leq$ `samplesize`:
   (a) `trials`++.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials=0, nevents=0, ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p=0`.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent++` and `ninteraction`$_i$`++`.
3. end while loop

Loop runs until number of scattering events (nevent) equal samplesize

- For each trial, nevent increments if at least one DM scatters inside the detector

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials=0, nevents=0, ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p=0`.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent++` and `ninteraction`$_i$`++`.
3. end while loop

Choose a production channel

$$\text{prob choosing ch. i} = \frac{\text{\# DM particles produced by channel i}}{\text{total \# DM particles produced}}$$

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials=0, nevents=0, ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p=0`.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to `output_file`, `nevent++` and `ninteraction`$_i$`++`.
3. end while loop

The Distribution corresponding to that production channel generates the 4-momenta of a meson/V from the p-target collision via rejection sampling using a differential particle production cross-section or by iterating through a pre-generated list of particle 4-momenta

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials`++.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

Append the initial meson/V to partlist

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to **output_file**, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

DMGenerator decays the initial V/meson, outputting a pair of dark matter particle 4-momenta.

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials=0, nevents=0, ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p=0`.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to `output_file`, `nevent++` and `ninteraction`$_i$`++`.
3. end while loop

If neither DM particle intersects the detector:

- choose another production channel

If either DM particle intersects the detector (the length of the intersection between the DM trajectories and the detector is nonzero):

- Determine whether the DM scatters.

If yes: an acceptance rejection algorithm is used to generate a final state 4-momentum by sampling from a differential distribution of interaction cross sections. Insert the end state Particle generated in partlist and set scatterswitch = True

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials=0, nevents=0, ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p=0`.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to `output_file`, `nevent++` and `ninteraction`$_i$`++`.
3. end while loop

If neither DM particle intersects the detector:

- choose another production channel

- Does this mean we move on to the next trial and choose another production channel (i.e. trials is incremented)?

- If not (during a trial, we choose production channels until the DM intersects the detector), how do we account for the (small) probability of a DM intersecting the detector?

Eg: in one run of BdNMC (Run 1499450926):

vnum[i] = # DM particles produced by this channel = 4.32 e12

 # DM intersecting detector = 1576

→ Will look at BdNMC code

If either DM particle intersects the detector (the length of the intersection between the DM trajectories and the detector is nonzero):

- Determine whether the DM scatters.

If yes: an acceptance rejection algorithm is used to generate a final state 4-momentum by sampling from a differential distribution of interaction cross sections. Insert the end state Particle generated in partlist and set scatterswitch = True

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials`++.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to `output_file`, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

If either particle scatters, then write all particles in partlist to the output_file, increment nevent, and increment ninteractions for the appropriate production channel

If neither particle scattered, then move on to the next trial

Cont...

- The total number of dark matter particles produced by each production channel is first calculated

1. Set `trials`=0, `nevents`=0, `ninteractions = zeros(chan_count)`.
2. While `nevent` $\leq$ `samplesize`:
   (a) `trials++`.
   (b) Generate a uniform random number `vrnd` $\in$ [0,`vnumtot`].
   (c) Set $i = 0$, `scatterswitch=False`.
   (d) While $i <$ `chan_count`:
      i. If `vrnd` $<$ `vnum`[i] then break
      ii. Else set `vrnd` = `vrnd` - `vnum`[i].
   (e) Initialize `Particle part` with four-momentum `p`=0.
   (f) Set the four-momentum of `part` using `Distribution`'s `Sample_Particle` method.
   (g) Initialize an empty `list` of `Particles partlist`.
   (h) Append `part` to `partlist`.
   (i) Generate a list of decay product `Particles` of `part` using `DMGenerator`$_i$'s `DMGen` member function and store them in `partlist`.
   (j) For each dark matter `Particle` j in `partlist`:
      i. If `Ldet`$_j$ == 0 then continue
      ii. Else
         A. Simulate an interaction using `Scatter`'s `probscatter` method.
         B. If probscatter returns true, then insert the end state `Particle` generated by `probscatter` after j in `partlist`, and set `scatterswitch=True`.
   (k) If `scatterswitch`, then write all the particles in `partlist` to `output_file`, `nevent`++ and `ninteraction`$_i$++.
3. end while loop

End the while loop when nevent = samplesize

At this point, the number of scattering from each production channel i is stored in ninteraction_i

How BdNMC calculates # of signal events

- For each production channel i:

$$\texttt{signal\_events}[i] = \frac{\texttt{ninteractions}[i]}{\texttt{trials}} \times \texttt{vnumtot} \times \texttt{pmax} \times \textbf{efficiency},$$

Note: x trials are done until samplesize total scattering events occur, where samplesize if an input parameter

- Sum of ninteractions[i] over all production channels = samplesize
- ninteractions[i] = number of scatterings by DM originating from production channel i

vnumtot is the total number of DM particles produced

- Pmax is “the maximum scattering probaility”
- Efficiency =detector efficiency
- Checked this formula by hand using the values for trials, vnumtot, etc outputed by BdNMC in terminal
- **Why do we multiply by pmax? Isn't the scattering probability taken care of in the trials?**

Reproducing figure 1.0 of "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniBooNE, T2K and SHiP":

- I run BdNMC ~20 times for m_V in [0.005, 0.75] for the production channels pi0 decay, eta decay, and proton bremsstrahlung (can't easily do parton production, but it doesn't matter for m_v<~0.75GeV) and signal channel NCE_nucleon.
- Parameters:

$m_\chi = 0.01$ GeV, $\epsilon = 10^{-3}$ and $\alpha' = 0.1$.

- Figure's caption says plot is for 9 GeV beam energy, but it's actually 8.9 GeV, they rounded.
- 2e20 POT
- Efficiency = 0.35
- 0.9 pi0_per_POT (pi0_per_POT is the number of pi0's expected per proton on target)
- Using production distributions pi0_sanfordwang for pi0 decay and k0_sanfordwang for eta decay (Sanford-Wang distributions appropriate for MiniBooNE energies)
- Production distribution proton_brem and ptmax = 0.2 , zmin = 0.3 , zmax = 0.7, which are appropriate for MiniBooNE
- Continued on next slide...

Reproducing some plots from the paper

- Want to reproduce figure 1.0 of "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniBooNE, T2K and SHiP":

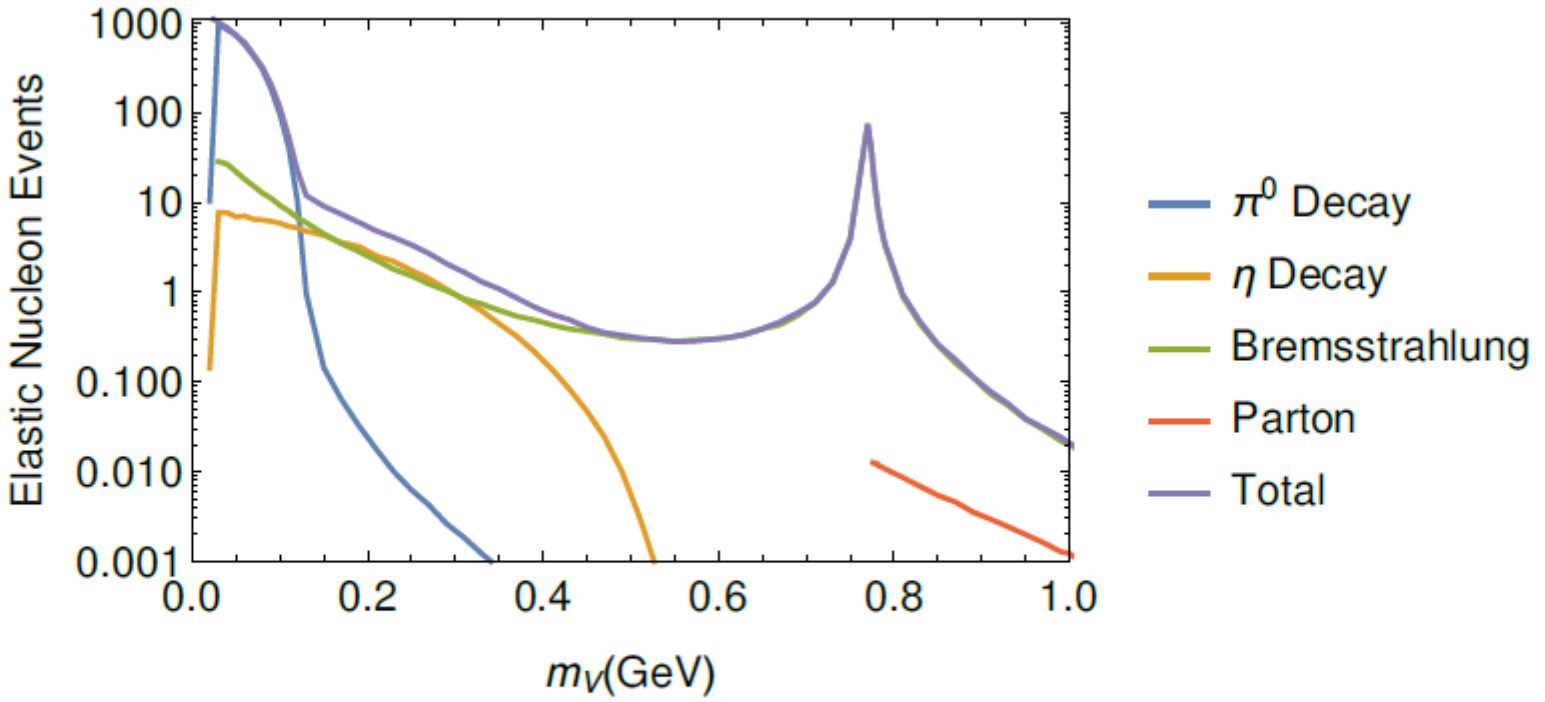


FIG. 1. A plot illustrating the distinct contributions to DM production (coupled through the vector portal), as discussed in the text, using the 9 GeV proton beam at MiniBooNE as an example. The rate of elastic scattering events on nucleons is plotted versus the vector mediator mass. From smaller to larger values of m_V, the dominant channels are π^0 decays, η decay, bremsstrahlung, which becomes resonant near the ρ/ω mass region, and finally direct parton-level production. The plot uses $m_\chi = 0.01$ GeV, $\epsilon = 10^{-3}$ and $\alpha' = 0.1$.

Reproducing figure 1.0 of "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniBooNE, T2K and SHiP":

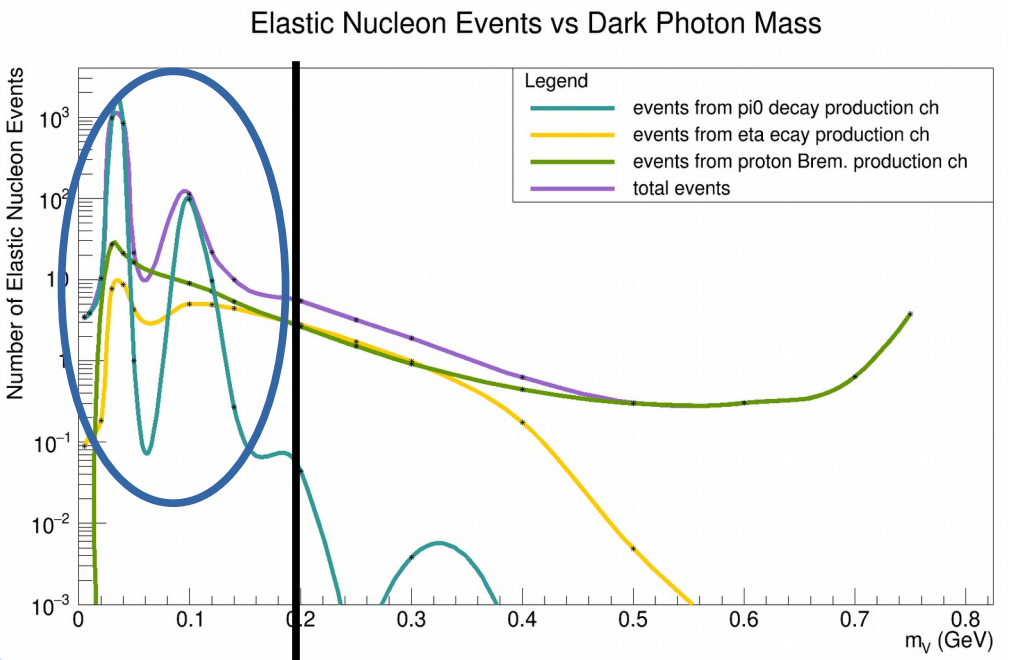
- Cuts on the kinetic energy of outgoing nucleon, default to min=0 and max=1e9 GeV:
- Initially Patrick said he used min=0.35 GeV and max=1 GeV
- Sensitivity section of paper says they use the same cuts (page 10)
- Using these cuts, my plot was off (~100 times fewer events than figure 1.0)
- I saw the paper "Dark Matter Search in a Proton Beam Dump with MiniBooNE" (arXiv:1702.02688v2) used 0.035 GeV as the lower cut in their analysis, do I used this as the lower cut
  - No longer off by a factor of 100, still a bit off.
  - Asked Patrick → 0.35 is a type, **it is actually 0.035GeV!** (upper energy cut is still 1.0 GeV)
- I'm also using the same value for the detector geometry as Patrick used (a bit different than values in the tables in paper):

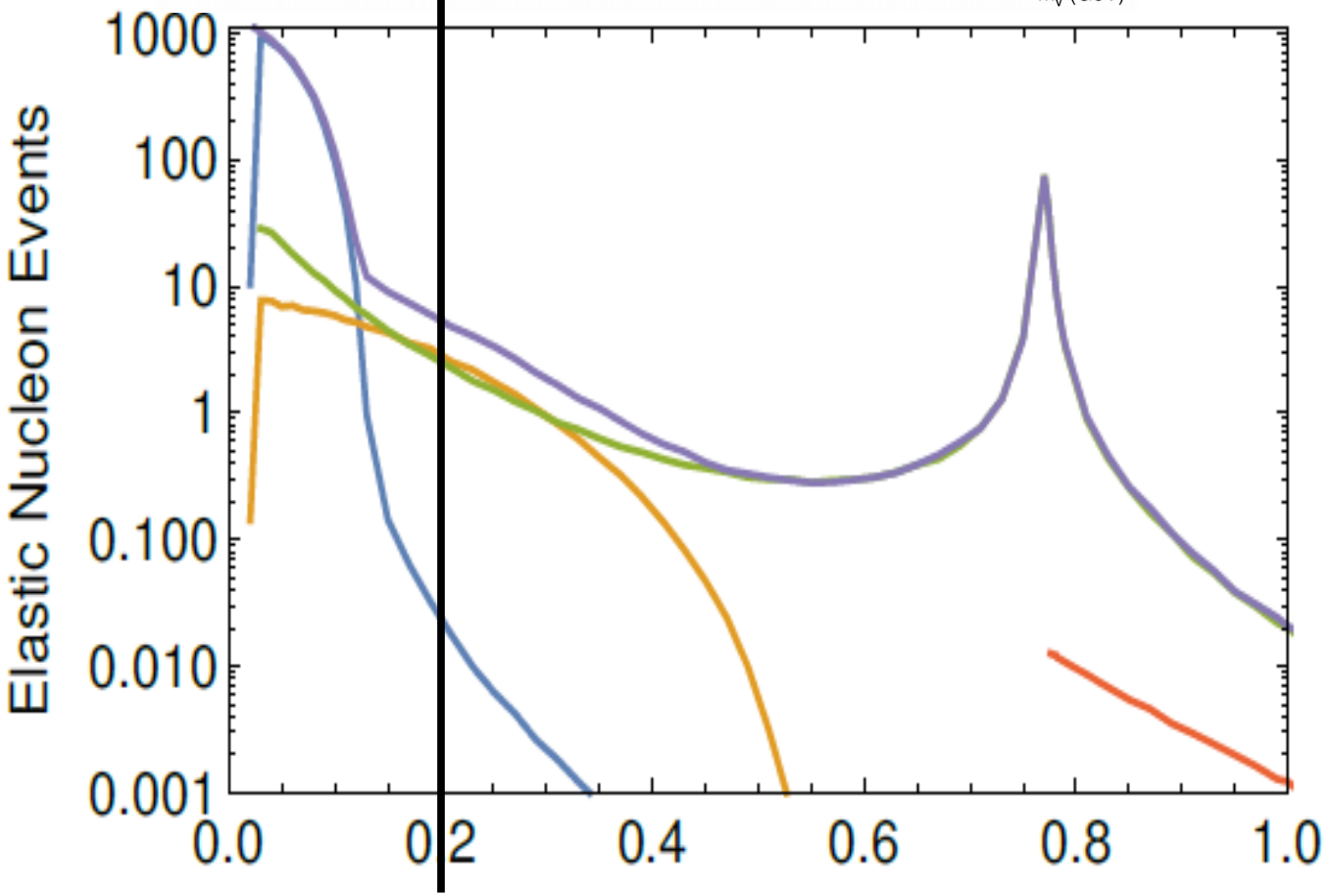
  x = 0.0 m

  y = -1.9 m

  z = 491 m

  r = 5 m
- My plot is on the next page (only 1000 entries for each BdNMC run, am currently running it with 20000 entries but that will take a few hours)





- Looks strange…
- Will look at BdNMC runs with more data first

Backup slides

A bit more info on the proton Bremsstrahlung production channel:

- User needs to state the values of these parameters when using the proton brem. channel:

```
#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
```

ptmax: The maximum transverse momentum which a produced V mediator may possess. The minimum is assumed to be 0.

zmin: The minimum value of $z = \frac{p_{V,z}}{P}$, where $p_{V,z}$ is the momentum of the V parallel to the z axis, and P is the total momentum of a beam proton incident on the target.

zmax: The maximum value of z, defined as in the **zmin**.

Calculating the total number of V bosons (N_V) produced by the proton bremsstrahlung production channel: (V mediators decay to dark matter isotropically)

$$N_V = \mathrm{POT} \int_0^{\mathrm{ptmax}^2} dp_\perp^2 \int_{\mathrm{zmin}}^{\mathrm{zmax}} dz \frac{d^2 N_V}{dz dp_\perp^2}, \tag{11}$$

Where the differential V production rate is:

$$\frac{d^2 N_V}{dz dp_\perp^2} = \frac{\sigma_{pA}(s')}{\sigma_{pA}(s)} F_{1,N}^2(q^2) w_{ba}(z, p_\perp^2), \tag{6}$$

where $s' = 2m_p(E_p - E_V)$, $s = 2m_p E_p$ and the photon splitting function is [54]

$$w_{ba}(z, p_\perp^2) = \frac{k_{V,B}^{(0)}}{2\pi H} \left[\frac{1 + (1-z)^2}{z} - 2z(1-z) \left(\frac{2m_p^2 + m_V^2}{H} - z^2 \frac{2m_p^4}{H^2} \right) + 2z(1-z)(z + (1-z)^2) \frac{m_p^2 m_V^2}{H^2} + 2z(1-z)^2 \frac{m_V^4}{H^2} \right],$$

with $H = p_\perp^2 + (1-z)m_V^2 + z^2 m_p^2$, and $k_{V,B}^{(n)}$ was defined above in (5). $\longrightarrow k_{V,B}^{(n)} = \begin{cases} \epsilon^2 \alpha (\alpha')^n & \text{for } U(1)' \\ \alpha_B^{n+1} & \text{for } U(1)_B \end{cases}$

Proton Bremsstrahlung Production Channel cont.

. The paper goes into detail about the "timelike form factor $F_{1,N}(q^2)$
(page 5 of "Light dark matter in neutrino beams: production modelling and scattering signatures at MiniBooNE, T2K and ShiP")

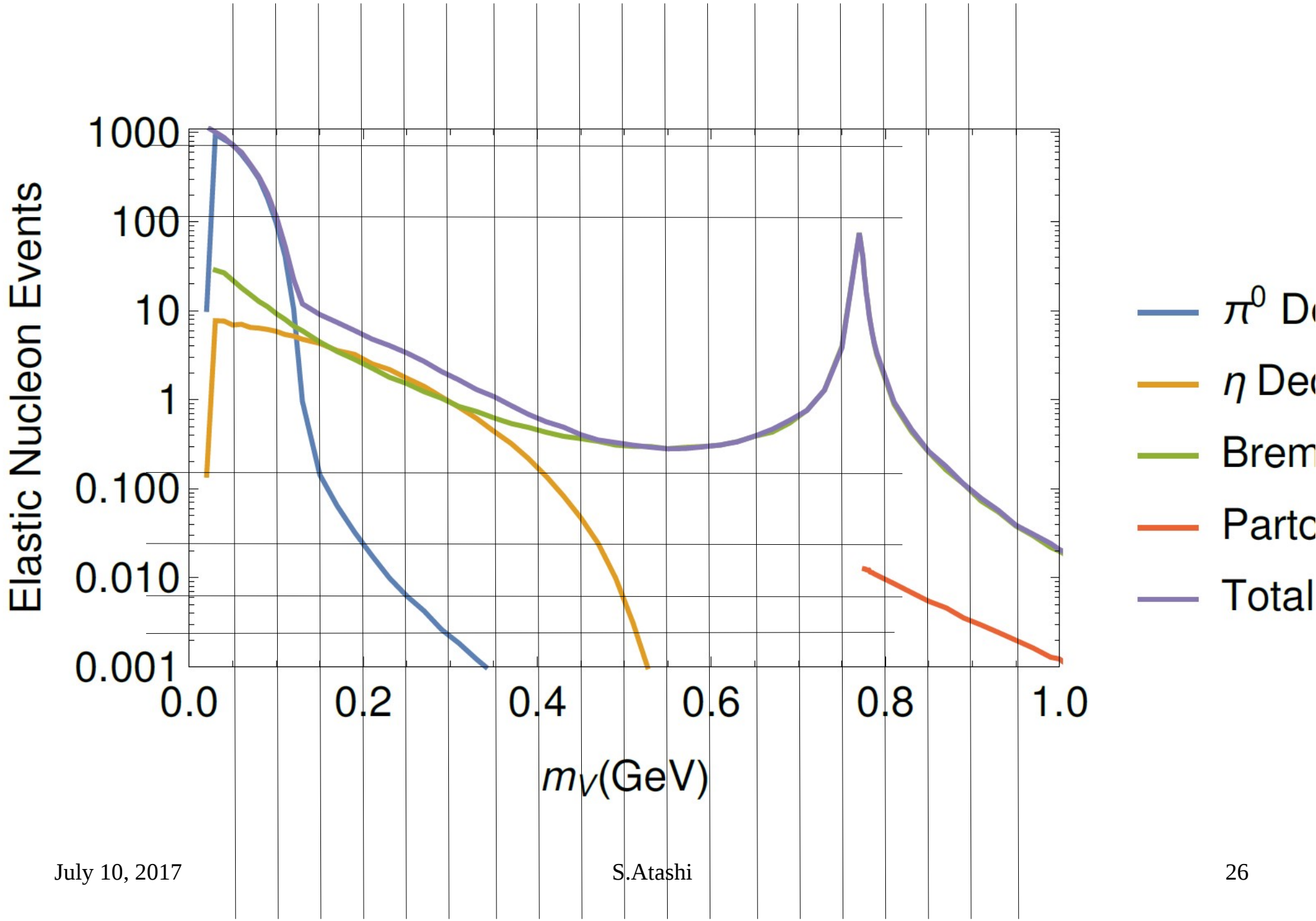
→ not really sure what they mean, but basically: the equation that the paper uses for the differential V production rate requires the following kinematic conditions:

$E_P >> m_P$

$E_V >> m_V$

$E_P - E_V >> |P_\perp|$

- So the range of values of z and $|P_\perp|$ over which the integral for N_V is calculated must be such that the conditions are met.
- Paper says these conditions are met if $z \in [0.1, 0.9]$ and $|P_\perp| < 1$ GeV for high energy experiments like SHiP, but not for lower energy experiments.
- eg for MiniBooNE (much lower energy than SHiP), the paper uses $z \in [0.3, 0.7]$ and $|P_\perp| < 0.2$ GeV (as in the parameter cards for the MiniBooNE-like experiments)


Elastic Nucleon Events
1000
100
10
1
0.100
0.010
0.001
0.0
0.2
0.4
0.6
0.8
1.0
m_V(GeV)
π^0 De
η Dec
Brem
Parto
Total