MC Overlay in NOvA

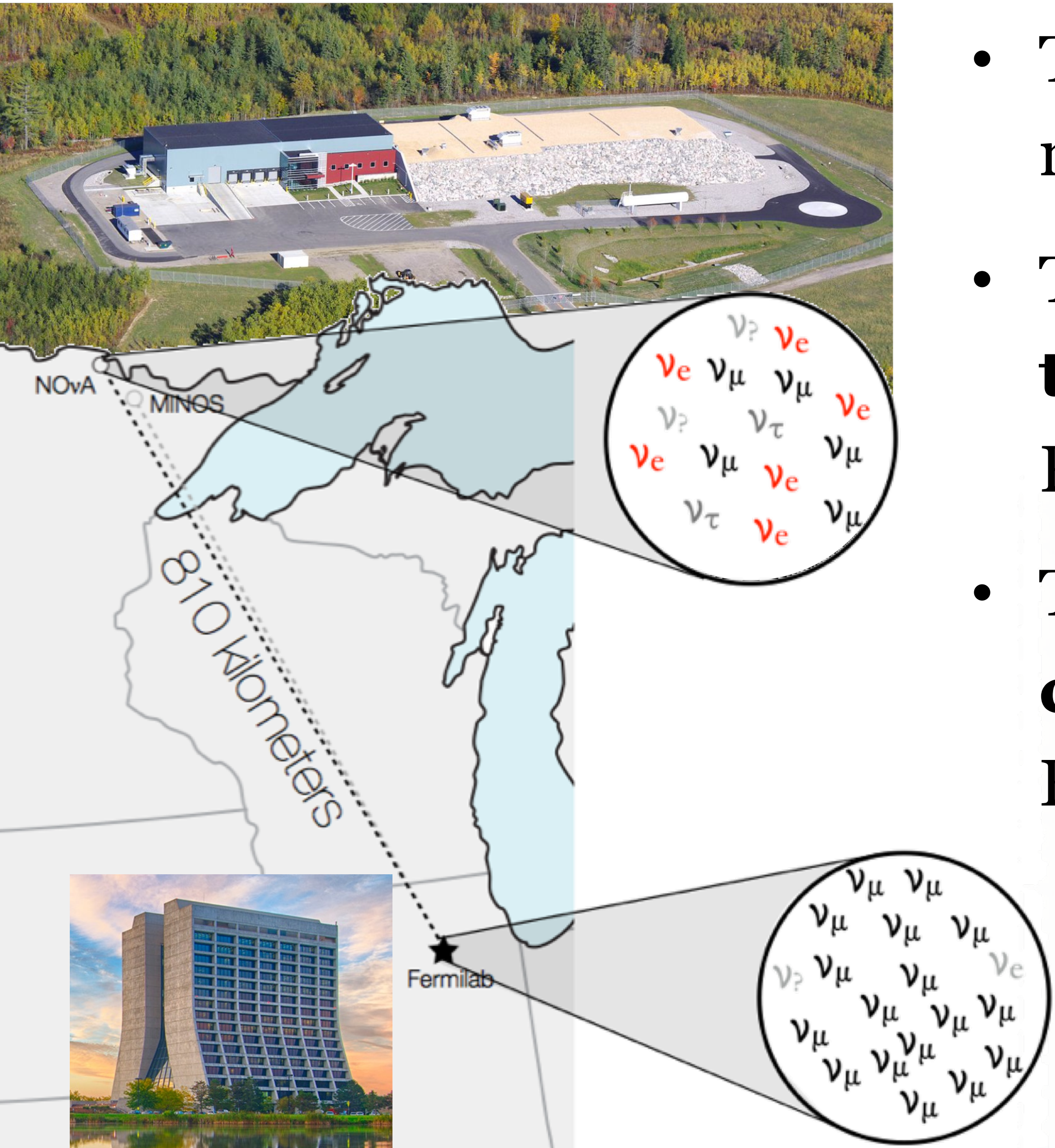
Alex Himmel, Fermilab

Introduction

- We use overlay for the same reason everyone here is thinking about it: our events often don't look like single neutrinos.

- In practice, we now use overlay in almost every MC sample we make.
- We use it to make some special samples, too, for specific studies.

The NOvA Experiment



- Long-baseline neutrino oscillation experiment.
- Two important things to note for this talk:
- The Far Detector sits **on the surface** at Ash River.
- The Near Detector sits **close to the target** at Fermilab.

Far Detector Overlay

- In the far detector we generate single neutrino events.
- We then overlay on that single neutrino event a real event from our cosmic triggered data.
- This accomplishes two goals:
  - We get the rate of overlap of neutrinos and cosmic background events correct.
  - We get the rate of stray hits sliced in with our neutrinos correct.
- Why not simulate the cosmics instead?
  - Our CRY simulation doesn't look much like our data.
  - And we have copious data to work with.

Far Detector Overlay Pitfalls

- It took us **many attempts** to get this right.
  - **Problem (1)**: We doubled the amount of detector noise since the data events also include that noise.
  - **Problem (2)**: You need to apply the data calibration to data hits and the MC calibration to MC hits.
  - **Problem (3)**: Not every data run has the same number of events in it.
    - What do you do if you run out of events in your overlay file?

Near Detector Overlay

- In the near detector, we generate a spill which generally contains several neutrinos.
- We then overlay onto that event “rock muons” – incoming particles into the detector from neutrino interactions in the rock.
- Why not just simulate the rock and detector neutrinos together?
  - Because rock simulation is **very** slow.
- Rock events are partially simulated separately as singles.
  - Need to front-load the inefficient Geant step.
  - But, we don’t do the final detector simulation so that the effect of multiple hits in a cell can be handled correctly.

Near Detector Overlay Pitfalls

- You might reasonably ask: if you're still simulating the rock events, where does the efficiency come from?
- The answer is also the key pitfall: resampling.
  - We re-use rock events 50-100 times.
- The key problem is if you have a rare event which passes selection cuts but has been resampled too often, you'll get weird spikes in distributions.
- How much resampling is too much?
  - You'll need to test to find out.

Special Samples

- Overlay also opens up interesting options for studying some tricky systematics.
- Our example:
  - We wanted to look for data-MC differences in “slicing”
  - This is the very first step of reco where we cluster hits together into neutrino candidates.
- Our overlay solution: overlay a single simulated neutrino onto both real and simulated beam spills.
  - Then you can see how the data vs. simulated environment might introduces difference in efficiency for that added neutrino.
- Also used for ν_e appearance at the near detectors in sterile searches.

Time-Varying MC

- Detector conditions change over time
  - We were already set up to simulate specific runs with "real" conditions in our MC.
- In the FD, when we randomly choose a run for the MC, we pull the corresponding cosmic file for that run to overlay.
- **Warning:** It takes care to get the sampling correct here.
  - Need to be aware of how POT, spills, and files interact.
  - This is important if beam intensity varies vs. time.
    - If you generate a fixed number of spills, and increase intensity, your file will contain more neutrinos and more POT.
  - If you don't get the above right, you will bias your sampling of low vs. high intensity.
- Our solution in the ND:
  - First randomly choose run numbers weighted by POT.
  - Then generate files with fixed POT/file, but potentially variable numbers of spills.
  - Ancillary benefit: reco time scales with number of neutrinos, not spills, so this also keeps downstream processing more predictable.

Technical & Production Considerations

- Be careful with module labels!
  - We put a lot of effort into making sure that the output of overlay looks just like the nominal generation so everything downstream will work transparently.
- Key assumption that overlay breaks: "MC generation is good for sites with slow transfer since that step doesn't need much input."
  - Our FD jobs require 1 ART file as input.
  - Our ND jobs require reading in a bunch of rock files to sample events from.
  - Adding in flux files, this means our **MC generation jobs require more input than our reco jobs**.

Technical & Production Considerations

- Overlay also generally requires significantly more complicated production scripting.
  - Our MC generation jobs take a fhicl file as input from a SAM dataset.
  - Figuring out all the other input files needed and fetching them happens in the production script layer.
- Also: avoid SAM queries inside your production jobs.
  - Our solution: we take snapshot, list the files in that snapshot, and put that file into a UPS product on CVMFS.
  - MC generation fhicls are given that dataset name and the filenames are pulled from that CVMFS location instead of from SAM.