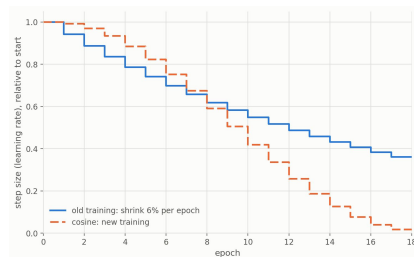


Attempted changes to existing DNN

- Wider network
 - Going from 16 to 32 neurons per layer in the network should allow for more subtle patterns to be recognized
- Vertex-weighted loss
 - Weigh the loss of points near the vertex strongly during training, because you shouldn't get as many points for guessing 0 at a random voxel as getting 1 at the true vertex
- Switching from Adam to AdamW optimizer
 - Optimizers decide how every learned parameter should change
 - AdamW pulls parameters slightly towards zero, which prevents some neurons from reacting very strongly to event-specific quirks
- Switching from constant 6% learning rate decay to cosine



Attempted changes

- Dropout
 - Turn off 10% of network's neurons at each training step, this sometimes helps other neurons learn backup clues that complement what it's already learned
- Gradient clipping
 - Cap the size of any single adjustment during training, so weird outlier events are less likely to screw up the network
- Dilation
 - Add 1 or 2 voxels surrounding charge-occupied voxels as potential locations.
- Combinations of the above.

Evaluation

13,216 BNB events 8,877 ν_e SBND production events

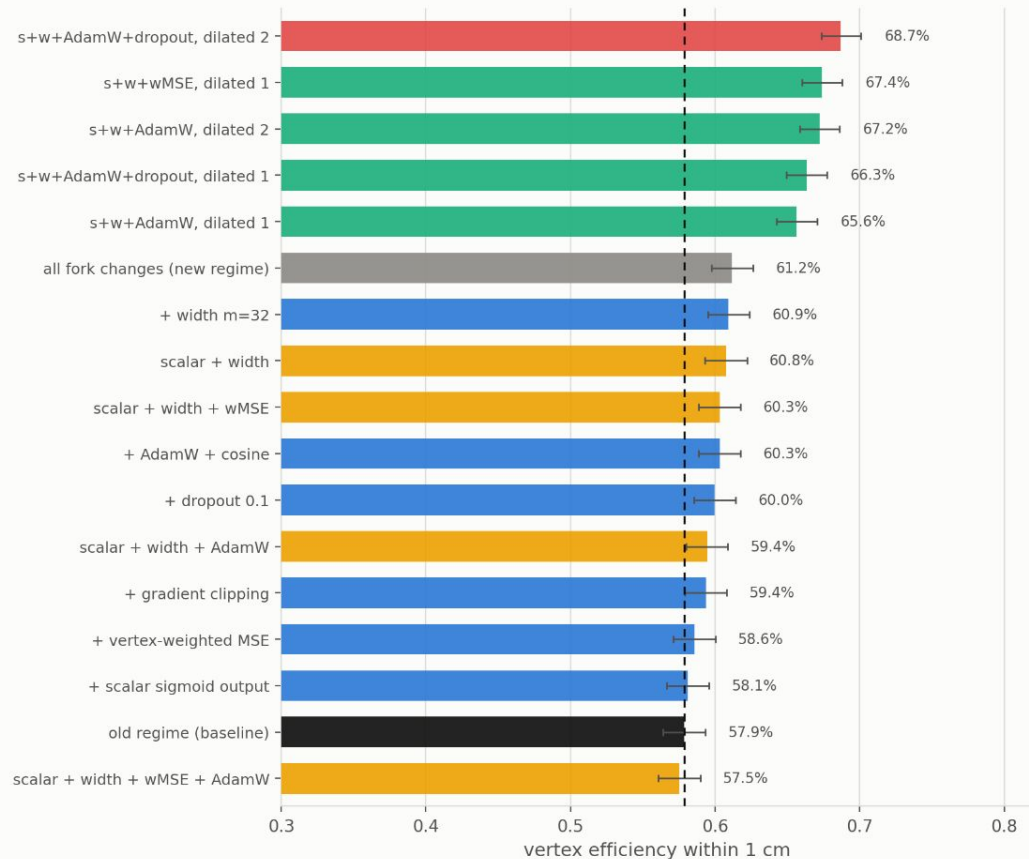
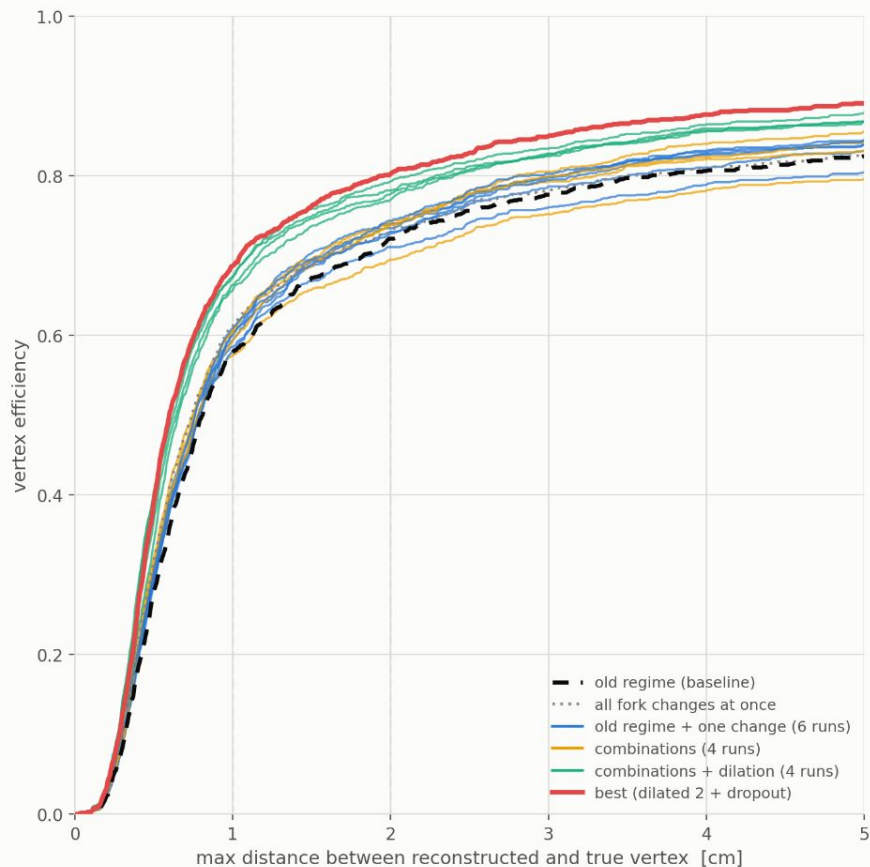
Conditions on events passed in:

- Neutrino inside active volume (lost 7,438 events)
- > 50 MeV deposited (lost 394 events)

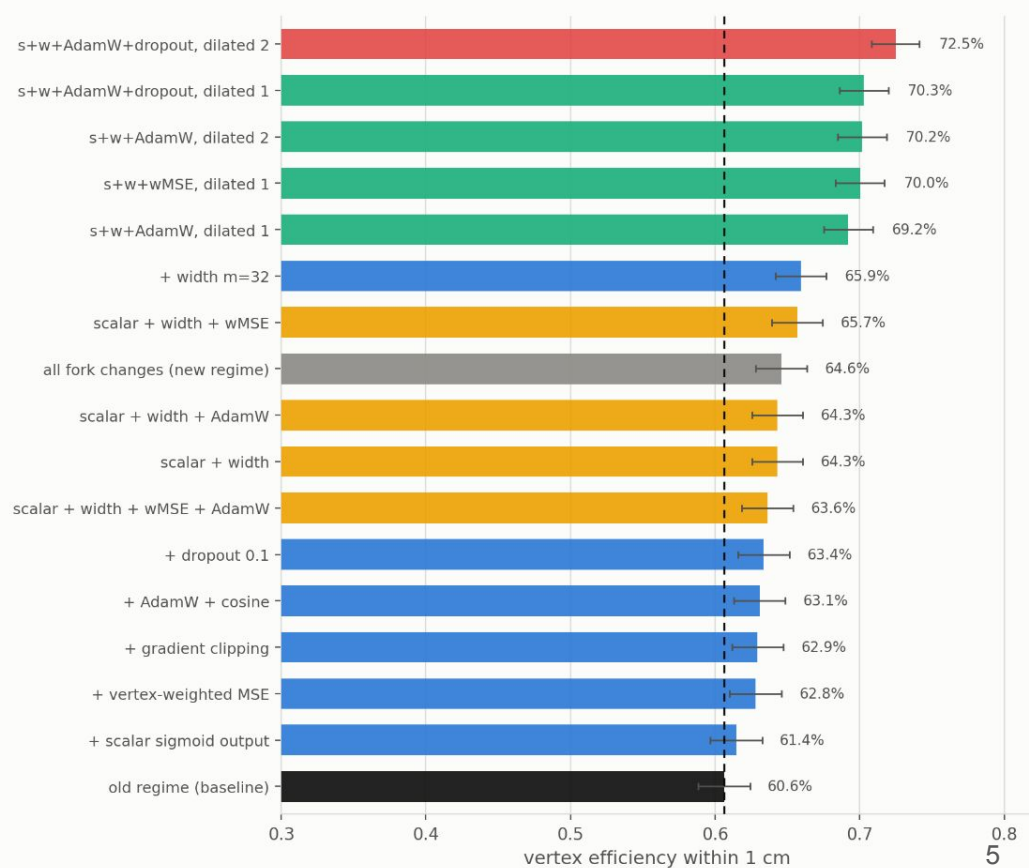
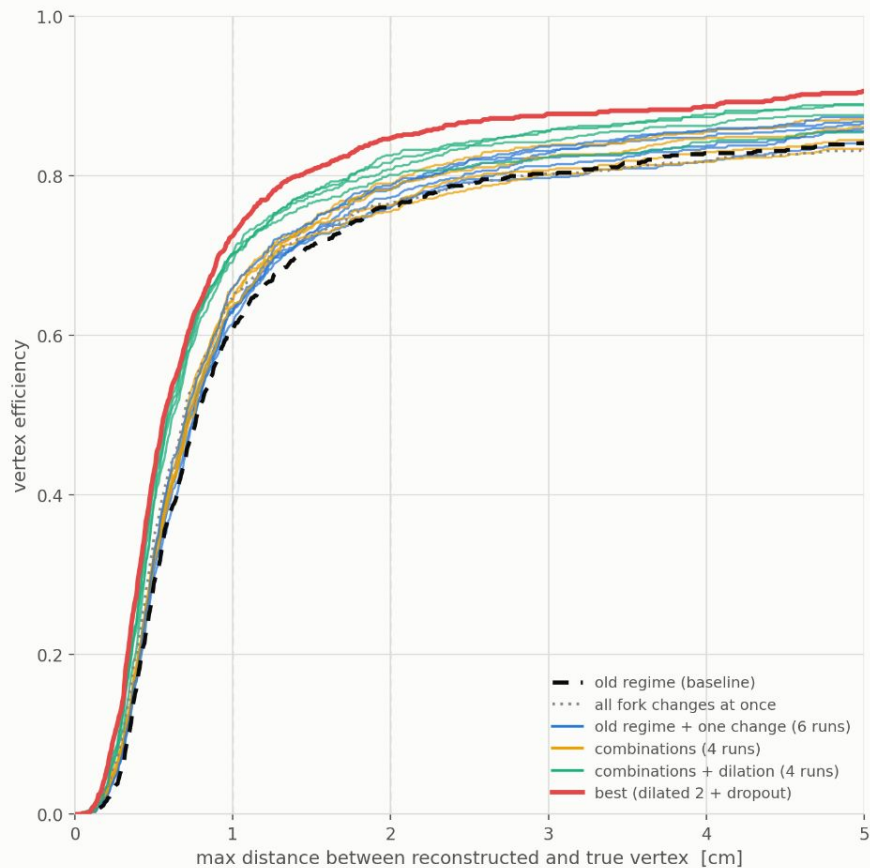
12093 events remaining. Trained on 10,035 events. 2188 events held out for evaluation. Remove 196 NC events for evaluation.

1972 events evaluated, 1311 ν_e CC, 661 ν_μ CC

nu_e



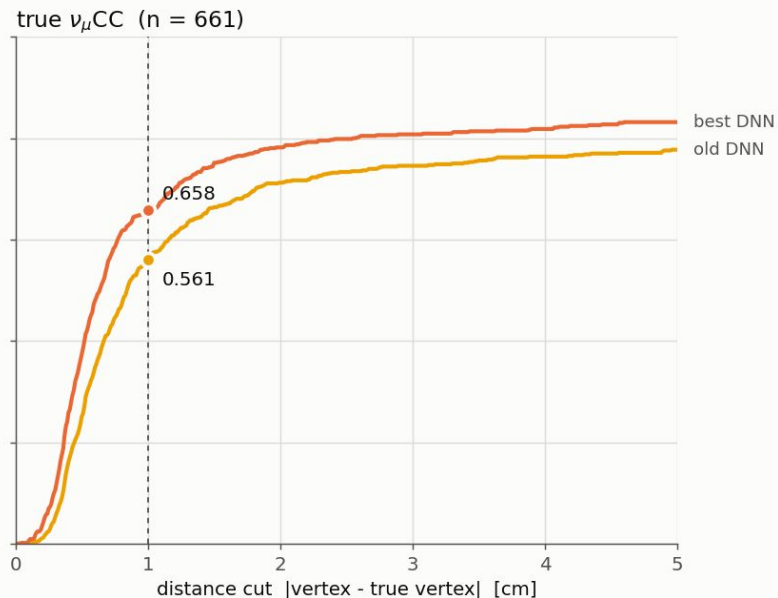
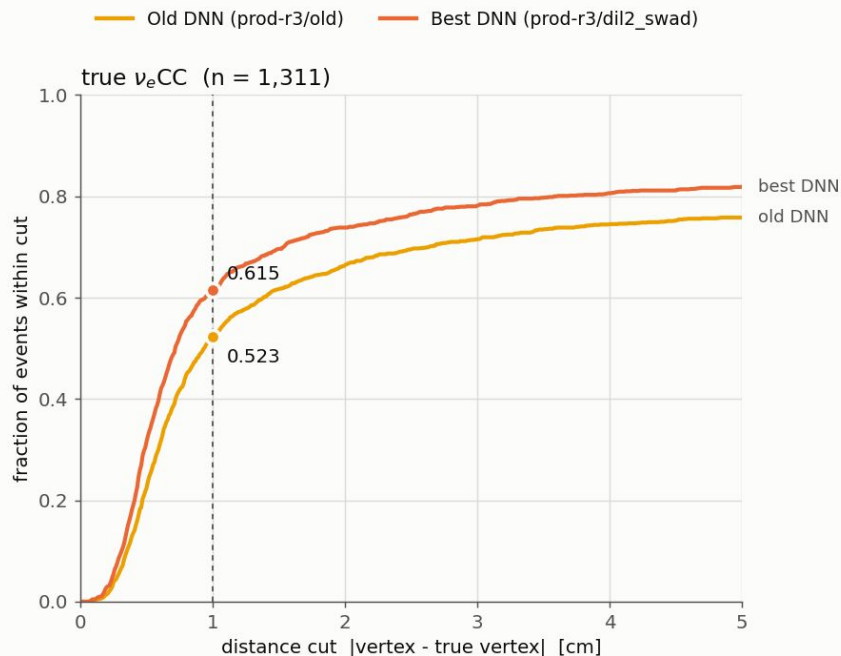
nu_mu



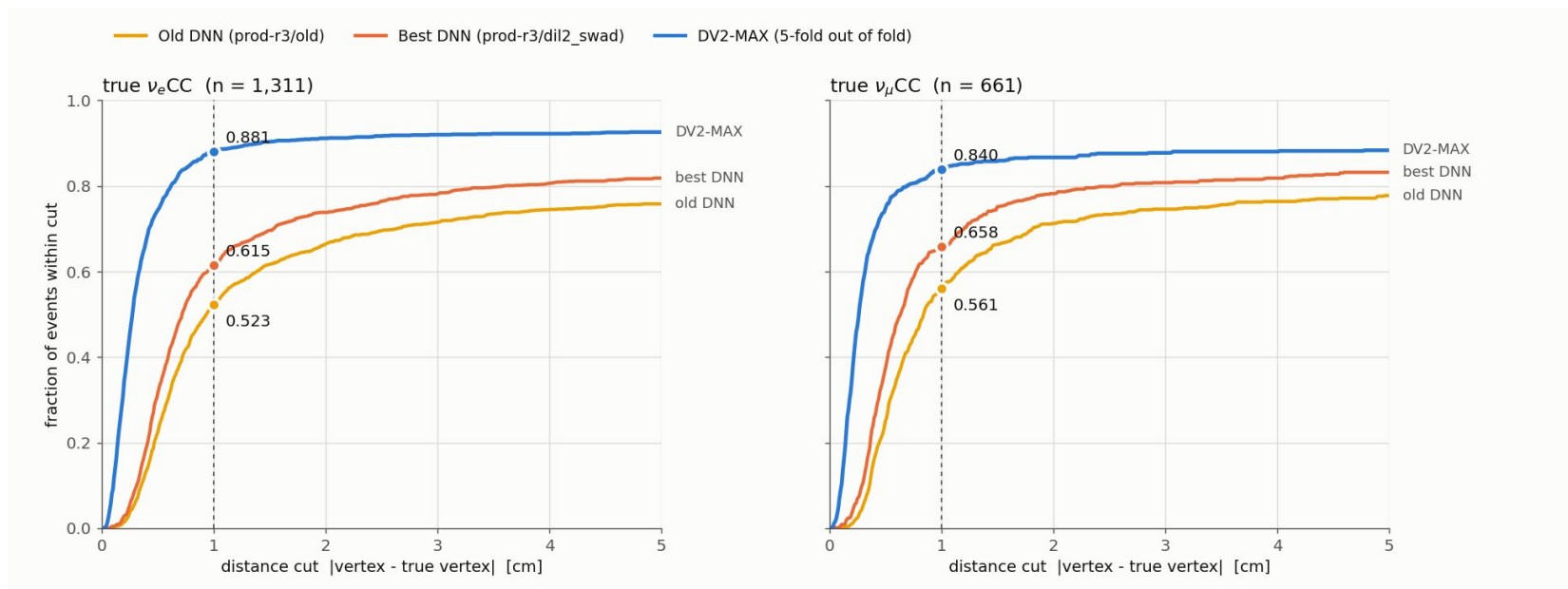
The best combination used wider network, AdamW, cosine decay, 0.1 dropout, 2-voxel dilation, but not weighted loss or gradient clipping

Flavor	Old DNN 1cm eff	Best Model 1cm eff
nu_e	52.3%	61.5% (+9.2)
nu_mu	56.1%	65.8% (+9.7)

Vast majority of this game comes from dilation, everything else only accounts for 2-3 points



Sparse CNN V2 Teaser



DeepVtx-2

- Similarly to DeepVtx, it's a sparse convolutional neural networks that voxelizes the detector and selects a voxel out of the ones that are charge-occupied

Why low efficiencies for old model

SBND's Wire-Cell reconstruction is worse at the vertex. The same traditional Wire-Cell vertex scores 0.418 on SBND ν_e CC against MicroBooNE's 0.507, and 0.560 against 0.713 on ν_μ CC. The DNN's input suffers too:

SBND MicroBooNE

Input has a point within 1 cm of the true vertex 0.80 0.86

No input point within 5 cm 7.7% 3.9%

Part of this is the x-shift export problem, about 3% of events.

The paper's DNN curve likely has the true vertex in its input (the `hybrid_vtx.py` voxel). That's worth about 3–4 points. Your leak-free MicroBooNE replication gets 0.630 for ν_e CC, against the paper's ~0.67–0.69.

Less training data. MicroBooNE trained on 48k ν_e CC events. SBND's old model trained on about 10k neutrinos, roughly 6k of them ν_e , which is about 3 doublings less. dlvtx's own SBND learning curves gain about 1–6 points per doubling.

Training/test input mismatch, in the in-chain evaluation only. The old model was trained on one truth-chosen cluster of exported fit points. The chain hands it every in-beam cluster, with differently sampled points. That costs about 7 points for choosing among clusters and about 4 for the sampling. MicroBooNE trained on its chain's own input, so it doesn't pay this.

Step 1: Build physics quantities

- Input: Space-points and charge
- Output: bragg peak, local geometries, wall distances, etc.

Step 2: Turn physics quantities into five outputs

- Input: Bragg peak, local geometries, wall distances, etc.
- Output: Heat (i.e. $P(\text{vertex within 3cm})$), any-nu weight, offset arrow to predicted vertex, $P(\text{vertex in cluster})$ $P(\text{event has a neutrino})$

Step 3: Voting [Pass 1]

- Each voxel uses its offset arrow to point to where it thinks the vertex is, and then weighs it by its any-nu weight
- Votes pile up into “modes” with a lot of votes
 - Modes within 10cm of a larger mode are merged into that larger mode
- 4 largest modes and 2 hottest voxels (peaks) are kept

Step 4: Zoomed-in neural network

- 64cmx64cmx64cm cube is generated around each mode/peak, with the mode/peak about between the center of the cube and the side opposite the drift direction
- Inputs are now physics quantities, and peak/mode heat, and a 2cm Gaussian blob at the peak/mode location
- For all voxels within the cube, outputs $P(\text{vertex within } 1\text{cm})$, offsets, $P(\text{vertex in cube})$

Step 5: Vote again [Pass 2]

- For every cube, perform the same voting procedure.
- Keep top 3 modes per cube

Model	Type	Idea in brief	Result (all / ve / BNB)	Status
DV2-MAX	Sparse CNN (DeepVtx successor)	Two-pass sparse 3D CNN; offset votes; LambdaMART picks the vertex	0.840 / 0.871 / 0.794	Best. All 5 folds done; leakage audit clean. v2 (pass 2 trained on displaced hints) is training now.
FOREST-MTT	Graph neural network	Dense points along tracks with physics features (dQ/dx, Bragg peaks); picks and nudges the vertex	0.798 / 0.821 / 0.764	Done, 5 folds; audit clean.
PHYSRANK-RZ	Non-transformer	Ranker + fine 0.2 cm CNN specialist	0.782 / – / –	Done, 5 folds. The stage-1 ranker alone scores 0.737. Fails its own strict go criterion: it loses 2.8% of stage 1's hits, against a 1.5% bar.
VertexFold	Transformer	AlphaFold-style refiner that moves the ranker's vertex	0.747 / 0.758 / 0.731	Done, 5 folds; audit clean. Starting it from FOREST-MTT's vertex gave no gain on 2 folds.
RENDER-SMC	Transformer	Renders the event from a guessed vertex; searches with particle filtering	Fold 0 only: 0.715 (DV2-MAX 0.841 on the same 2,847 events)	Folds 0–1 trained (validation 0.728 / 0.722). Fold 1 inference is queued (191685). Folds 2–4 were never trained.
GRV	Transformer	Generate many candidates, refine, verify	0.642 vs ranker 0.609, on 800 dev events only	Not comparable to the rest. The full run was never started; only the control test (191390) was run.
HIVE-DETR	Transformer	DETR-style query decoder	0.558 (its control: 0.608)	Parked. Only 2 of 5 folds ever learned (0.656, 0.682), and turning off GO-LSD didn't fix the stuck runs.
Only DV2-MAX, FOREST-MTT, PHYSRANK-RZ and VertexFold have full 5-fold results on the same 14,261 events. RENDER-SMC's number is from one fold, and GRV's is from a small dev set.				

Step 6: Collect candidates

- Collect all modes from pass 1, pass 2 and wirecell traditional, and all voxels within 3cm of top pass 1/pass 2 modes
- Each candidate has 340 features describing, charge, nearby wirecell objects, and 40 cnn features, like distance to each cnn mode, mode scores and ranks, heat and vertexness, $p(\text{host cluster})$