awesome — I took your rough draft and turned it into a polished, research-facing slide deck outline that cites your codebases directly and recommends figures you can drop into the slides. I kept the AI talk light and the biology angle front-and-center.

**Slide-by-slide draft (titles, bullets, notes, figure ideas)**

**1) Title**

**Predicting B-cell conformational epitopes to accelerate antigen design**  
Presenter: *[Your Name]* • Vaccine and Infectious Disease Organization (VIDO)

*Notes:* Frame the talk as shortening wet-lab iteration in antigen design.

**2) Objective — why this matters**

* Rapidly **prioritize epitope-rich antigen regions** before assays
* Reduce **trial-and-error** and cost in antibody discovery and vaccine design
* Focus: **antigen–antibody interface (conformational epitopes)**, not linear peptides

*Figure:* simple cartoon of an antibody docking on a protein surface.

**3) Problem framing: antigen–epitope interface prediction**

* Conformational epitopes are **3D and discontinuous** → require structure-aware signals
* We predict, **per residue**, the probability of belonging to the antibody–antigen interface

*Notes:* Use one sentence to distinguish “linear epitope” vs “conformational epitope”.

**4) Data: source and curation**

* Source: **SAbDab** entries with single protein antigen chains
* **Filtering & dedup**: drop entries with multiple antigen chains; deduplicate by PDB ID (keeps first) (code: filter\_and\_deduplicate\_tsv)
* **Define positives (labels):** residues within **6 Å** of antibody atoms in the solved complex (heavy/light chains) (code: identify\_epitope\_residues)
* **Homology-aware splits:** cluster antigen sequences by **CD-HIT at 40% identity**, cap cluster size, then split **by cluster** → avoids train/test leakage (pipeline in sequence\_clustering.py)
  + Configured via CDHIT\_THRESHOLD = 0.4 and SPLITS\_FILE\_PATH
* **External test (domain shift):** *All SARS-CoV-2 Spike (S) complexes were excluded from training/validation and used* ***as the only test set*** *to assess generalization to a novel antigen family.*

*Figure(s):*  
A. Flow diagram: SAbDab → filtering/dedup → PDB cleaning → **CD-HIT (40%)** → cluster-level train/val/test split.  
B. A contact-map sketch showing 6 Å interface labeling.

**5) Existing methods & their limitations (baseline context)**

* **DiscoTope**: structure-based, but published data pipelines can **leak homologs across splits** if clustering isn’t enforced; we **re-trained/re-evaluated** on our homology-clean splits for a fair baseline.
* **BepiPred**: sequence-only → **no 3D context** (limited for conformational epitopes).

*Notes:* Keep this as motivation for your pipeline rather than a teardown.

**6) Features/signals our pipeline uses (biologist-friendly)**

* **Surface exposure** *(RSA from SASA)* and **B-factor** per residue (get\_biophysical\_features)
* **Sequence context:** per-residue **ESM-2** embeddings (up to 1022 aa; model esm2\_t33\_650M\_UR50D)
* **Structure-aware context:** **ESM-IF1** inverse-folding encoder features from antigen coordinates
* **Amino-acid identity** (one-hot) and chain/length metadata assembled into X\_arr per protein (see feature\_engineering.py / structure\_data\_to\_dict)

*Figure:* “feature stack” graphic: RSA/B-factor + ESM-2 + ESM-IF1 → concatenated per residue.

**7) Model v1 (sanity-check): XGBoost**

* **GroupKFold (by protein)** to prevent residue-level leakage; **class imbalance** handled via scale\_pos\_weight; **5-fold CV**; early stopping on validation (AUC-PR) (train\_cv)
* Best CV model saved; **optimal trees discovered** then **final model retrained** on all train+val and saved to models/final\_model.json
* **Operating point:** default display at PREDICTION\_THRESHOLD = 0.6 for confusion matrix & summaries (config)

*Figure(s):*  
A. CV schema (grouped by PDB).  
B. Small box showing “Optimum trees = 2999” if you keep that result; it’s printed in your pipeline output.

**8) Results — External Spike-only test**

The metrics shown here are **exclusively** from the held-out **SARS-CoV-2 Spike** set to probe domain-shift generalization.

**Re-trained DiscoTope on our splits:**

* AUC-ROC **0.7788**; AUC-PR **0.1234** *(on our hold-out)*

**Our XGBoost (v1):**

* **Cross-val (5-fold):** AUC-PR **0.5195 ± 0.0056**; AUC-ROC **0.8841 ± 0.0021**
* **Final test:** AUC-PR **0.2524**; AUC-ROC **0.8411** (from final evaluation block)

*Explain briefly:* AUC-PR is stricter under heavy imbalance; AUC-ROC complements it.

*Figure(s):*  
A. **Precision–Recall curve** saved by save\_precision\_recall\_curve to evaluation\_results/pr\_curve.png. Drop that image into the slide.   
B. **Confusion matrix** at threshold 0.6 (evaluation\_results/confusion\_matrix.png).

**Slide 9 — Case study: SARS-CoV-2 Spike RBD + S309/S2X35 (PDB: 7R6W)**

**Bullets (swap in your specific PDB ID, chain, and numbers):**

* **Design:** Train on non-Spike; run inference on **[PDB:XXXX, chain A, RBD/NTD as noted]**.
* **Qualitative check:** Predicted epitope residues **cluster at known Ab interface** (e.g., RBD ridge/NTD supersite).
* **Threshold behavior:** Increasing the decision threshold prunes diffuse predictions while **retaining core interface residues**.
* **Numbers @ τ=0.6:** Precision **\_\_%**, Recall **\_\_%**, F1 **\_\_** (per-protein).
* **Case-study bullet:** “At τ=0.6 on **[PDB:XXXX]**, **\_\_%** of predicted residues fall on the known Ab-antigen interface; raising τ to 0.9 retains the core footprint while reducing off-site predictions.”

**Figure layout (3 panels, left → right):**

1. **Color by probability (0–1 gradient)** on one protomer (cartoon); show glycans as sticks if present.
2. **Threshold sweep** overlays @ τ=0.3 / 0.6 / 0.9 (three small snapshots or one montage).
3. **Per-protein PR curve** with the chosen operating point marked.

Speaker note: “This is the exact complex: **[PDB:XXXX]**. You can see high-probability residues sitting at the antibody footprint; at higher thresholds we retain the core interface.”

**What this gives you**

* A heat-colored protomer (blue→red = low→high probability).
* Semi-transparent surfaces for τ=0.3 / 0.6 / 0.9 so you can export a 3-panel threshold montage.
* Ground-truth epitope residues in **green sticks** for quick TP/FP/FN eyeballing.

**Simple per-protein PR curve (numbers to quote on the slide)**

From your evaluation artifacts, plot the **PR curve for this PDB** and mark the threshold τ you used for the visualization (e.g., τ=0.6). If you don’t already export per-protein PR curves, compute precision/recall by sweeping τ over [0..1] on just the residues of this PDB and save a small PNG.

**Suggested caption to paste below the figure**  
“**[PDB:XXXX, chain A]** — Residue-level epitope probabilities (blue→red). Surfaces show predictions at τ=0.3/0.6/0.9. Green sticks = true interface residues (6 Å). Core interface remains at high τ, indicating robust signal.”

**10) What’s different about our approach**

* **Leakage-resistant splits** via **CD-HIT 40%** clustering and cluster-level split (vs naïve random)
* **Hybrid signals:** RSA/B-factor + **ESM-2** (sequence) + **ESM-IF1** (structure) per residue
* **Imbalance-aware training** and **grouped CV**; report both **AUC-PR** and **AUC-ROC**
* **Domain-shift evaluation** on **Spike-only** external test.

*Figure:* side-by-side “baseline vs ours” schematic.

**11) Roadmap: larger transformer (production model)**

* Plan: train a **transformer for tabular/per-residue features** (ExcelFormer backbone) with **mixup**, **cosine LR**, **early stopping**, **AUC-ROC** metric (already scaffolded in your repo)
* **Optuna tuning** over layers/heads/token dim, LR & weight decay; mixup type/β also tuned (scripts: tune\_only\_mix.py, tune\_fully\_after\_mix\_tune.py)

*Figure(s):*  
A. Model block diagram (input features → transformer → per-residue probability).  
B. Hyperparameter search grid cartoon.

**12) Data improvements & augmentation plan**

* **PTM-aware negatives:** add features for **glycosylation/phosphorylation** masking; residues under bulky PTMs often **less likely** epitopes (will encode as features and/or priors). For example, Spike is heavily **glycosylated**, PTM-aware features are planned.
* **RFdiffusion-based scaffolding for motif augmentation:**
  1. **Extract known epitope motif** (e.g., residues 50–65 → epitope\_A\_motif.pdb)
  2. **“Scaffold” 100-residue mini-proteins** that present the motif (generate 10 designs)
  3. **Label** motif residues = 1, scaffold residues = 0, and use for **pre-training**, not for final test  
     *Caution:* keep synthetic data **out of the evaluation** to avoid distribution shift.

*Figure:* 3-panel schematic: motif extraction → RFdiffusion design set → labeled training pairs.

**13) How we’ll validate biologically**

* **Threshold calibration** to match desired **precision or recall** for triage (tune from PREDICTION\_THRESHOLD=0.6)
* **Wet-lab validation:** pick **top-ranked residues** for mutagenesis/ELISA/competition assays; verify loss of binding when mutated.
* **Generalization checks:** per-protein metrics and **held-out clusters** to ensure robustness.

**14) Impact for VIDO + practical next steps**

* **Use cases:** antigen construct design, epitope-focused immunogen engineering, down-selection for structural studies
* **What we need:**
  + GPU access for transformer training (PyTorch, Torch ≥2.6) and ESM embeddings
  + PTM annotation sources (glycan maps)
  + Occasional structural curation support
* **3-month plan (proposal):**
  + Month 1: finalize PTM features + retrain XGB; establish wet-lab shortlist
  + Month 2: transformer tuning + ablations; interim review
  + Month 3: prospective validation on **new antigens**

**15) Backup: methods details (for Q&A)**

* **Labeling details** (6 Å, heavy/light chains) and RSA/B-factor computation (Shrake-Rupley)
* **Files/paths:** STRUCTURED\_DATA\_PATH, SPLITS\_FILE\_PATH, evaluation outputs (pr\_curve.png, confusion\_matrix.png)
* **Feature importance:** XGBoost feature importance plotting utility is included (nice to have a bar chart slide)

**Figures & where to get them (quick wins)**

* **PR curve** and **confusion matrix**: auto-saved by evaluate.py to evaluation\_results/ (save\_precision\_recall\_curve / save\_confusion\_matrix) — drop those PNGs straight into slides.
* **Top features bar chart**: run your plot\_feature\_importance helper on the final model to export a bar plot (suggest top\_n=30).
* **Per-protein metrics**: per\_protein\_metrics.csv is written by evaluate.py; plot median/variance across proteins for a backup slide.
* **One-protein “report” table**: predict\_standalone.py and related utilities print/CSV predict scores & top residues; screenshot or export as table.

**Polished wording you can lift into slides**

**Objective (slide 2):**  
“Identify conformational B-cell epitopes directly on antigen structures so we can prioritize constructs and mutations before going to the bench.”

**Data & labels (slide 4):**  
“We curate single-chain protein antigens from SAbDab, deduplicate PDB entries, and define epitope residues as those within 6 Å of antibody atoms in the co-crystal. To prevent inflated performance from homologs, we cluster antigens at 40% identity (CD-HIT) and split by cluster.”

**Signals (slide 6):**  
“We combine surface exposure (RSA), flexibility (B-factor), amino-acid identity, and context from two protein language models: ESM-2 (sequence) and ESM-IF1 (structure-aware).”

**Results (slide 8):**  
“On our homology-clean test set, XGBoost outperforms a re-trained DiscoTope baseline (AUC-ROC 0.841 vs 0.779; AUC-PR 0.252 vs 0.123). Cross-validation AUC-PR averages 0.520, reflecting strong precision at practical recall.”

**Roadmap (slide 11):**  
“Next, we’ll train a transformer purpose-built for per-residue tabular features with mixup regularization and Optuna tuning. This should capture longer-range dependencies better than boosted trees.”

**Final suggestions**

* Keep metrics **biologist-friendly**: explain AUC-PR in one sentence and show an **operating-point** example (e.g., at 60% precision, what recall do we get?).
* Add one **case study** slide: show a known antigen where your top 10 residues include many true interface residues (use your single-protein inspection utility).
* Close with an **ask + timeline** slide (resources, PTM annotations, wet-lab validations).

If you want, I can also turn this outline into a clean, branded slide deck template (titles, body text, placeholders, and figure boxes) and hand it to you as a .pptx—just say the word and I’ll generate it with your VIDO colors/logo.