Hybrid blueprint (CNN14 ⊕ PaSST)

Shared front-end (both datasets)

- **Resample**: use the native SR (CirCor ≈ **4 kHz** from .hea; Yaseen can stay 22.05 kHz or be downsampled—see per-dataset notes).
- **Window**: fixed clip length (CirCor **12–12.5 s**; Yaseen **3 s**). Pad or crop as needed.
- **(Optional) DWT denoise**: **db4, level-5**; soft-threshold detail coeffs (BayesShrink/VisuShrink). Apply on waveform, then re-pad.
- **Time–frequency**: 128-mel, 25 ms win / 10 ms hop → **log-mel** → per-sample **z-score**. Resize/crop to **224×224** for PaSST/Vision backbones.
- **Aug (train-only)**:
    - Waveform: small Gaussian noise, ±10% time-stretch, random time-shift.
    - SpecAugment: 1–2 time masks & 1–2 freq masks (≤10% each).

Tensor to encoders (per site/clip): X ∈ (B, 1, 224, 224) (treat log-mel as 1-channel image).

Dual encoders (shared across sites)

- **CNN branch**: **PANNs CNN14** (AudioSet-pretrained), first conv adapted to 1-ch. Output pooled embedding **e_cnn** $\in \mathbb{R}^{1024}$.
- **Transformer branch**: **PaSST-S** (Patchout Spectrogram Transformer) on 224×224. Output pooled embedding **e_tr** $\in \mathbb{R}^{768\text{–}1024}$ (model-dependent).

Site embedding: [e_cnn ⊕ e_tr] → LN → Dropout → Linear(…→512) → **z_site** $\in \mathbb{R}^{512}$.

CirCor-2022 pipeline (multi-location, multi-task)

A) Data packaging

- **Group by patient**; collect sites **{AV, MV, PV, TV, (Phc)}** available for that patient.
- **Split**: **StratifiedGroupKFold (5-fold)** by patient. Stratify primarily on **murmur** (Present/Absent/Unknown) or **outcome** (Normal/Abnormal); secondarily maintain **campaign (CC2014/2015)** and **age-group** balance.

B) Per-site processing

For each site recording:

1. resample → window (12–12.5 s) → **DWT (optional)** → log-mel(224×224) → z-score → **augment** (train-only) → to encoders → **z_site**.

C) Multi-location fusion (patient level)

- Add a **learned location embedding** to each z_site (one vector per site code).
- **Attention pooling (MIL)** across sites → **z_patient ∈ $\mathbb{R}^{512}$**.
  (If a site is missing, skip it; optionally include a learned NULL vector for padding.)

D) Heads & losses

- **Head-A (Murmur)**: 3-class softmax {present, absent, unknown}.
  Loss: **class-weighted CE** (e.g., Present and Unknown up-weighted).
- **Head-B (Outcome)**: sigmoid {abnormal vs normal}.
  Loss: **BCE**.
- **Total**: L = α·CE_murmur + β·BCE_outcome (start **α=0.6**, **β=0.4**).
- **Sampling**: patient-level batches; ensure label/campaign/age mix per batch.

E) Training schedule

- **Optimizer**: AdamW, lr 2e-4, cosine decay, warmup 5 epochs, AMP on.
- **Freeze/unfreeze**: first 5–10 epochs freeze CNN14 lower blocks and PaSST patch embed → then unfreeze all.
- **Early stop** on **murmur weighted-accuracy**; also track **outcome AUROC/AUPRC**.
- **Metrics** (match Challenge): **weighted-accuracy (murmur)**, **cost-based (outcome)** + AUROC/AUPRC/confusion matrices.

F) Inference

- Per patient: encode all available sites → attention pool → output **both heads**.
  Optionally average over multiple windows per site and then pool across sites.

Yaseen pipeline (single-recording, 5-class)

A) Data packaging

- **Per file** (usually single-site): keep dataset native SR (**22.05 kHz**) or **downsample to 4 kHz** to reuse CirCor front-end; choose one and stick with it for consistency.
- **Split**: **StratifiedKFold (5-fold)** on the **5 classes** {Normal, AS, MS, MR, MVP}.

B) Per-record processing

- window **3 s** → **DWT (optional, db4 level-5)** → log-mel(224×224) → z-score → **augment** (train-only) → encoders → **z_site** (there's just one).

C) Classification head

- **Head-Y (Disease-5)**: softmax over {Normal, AS, MS, MR, MVP}.
  Loss: **CE** (+ class weights if imbalanced).

D) Training schedule

- Same optimizer/schedule as CirCor; early stop on **macro-F1** or **balanced accuracy**.

E) Inference

- Single clip → dual encoders → project to **z_site** → **Head-Y**.

What DWT buys you (and where to add it)

- **Where**: immediately after resampling and before TF features.
- **Why**: removes high-frequency noise while preserving heart-sound transients (S1/S2 and murmur bands).
- **Recipe**: x_denoised = wavedec(x, 'db4', level=5) → soft-threshold detail coeffs (BayesShrink) → waverec → pad/trim → TF.
- **Ablate**: keep a toggle; on noisier recordings (field recordings, infants), DWT usually gives a small but reliable lift.

Shapes & modules (quick wiring)

- **Input (per site)**: (B, 1, 224, 224) log-mel
- **CNN14 →** GAP → (B,1024)
- **PaSST →** class token / mean pool → (B,768–1024)
- **Concat →** (B, ~1800–2000) → LN → Dropout → Linear → **(B,512)** = z_site
- **MIL attention over sites** → **z_patient (B,512)**
- **Heads**: Murmur (B,3), Outcome (B,1); Yaseen (B,5)

Training knobs that matter

- **SpecAugment intensity**: keep masks small (≤10%) to avoid deleting murmurs.
- **Class weights (murmur)**: tune on fold-0 using val weighted-accuracy.

- **Patchout (PaSST)**: enable light patch dropping to regularize.
- **Demographics (CirCor)**: optional—concat a small MLP on age group & campaign and add to z_patient.

What to log

- Losses per head; murmur WA; outcome AUROC/AUPRC; confusion matrices; site attention weights (to verify MV/AV often dominate when murmurs are present); DWT on/off ablation; CNN-only vs PaSST-only vs **hybrid**.

If you'd like, I can turn this into a minimal PyTorch skeleton (Dataset + PaSST & CNN14 encoders + MIL attention + two heads, with a DWT toggle and ready scoring stubs). It will follow the exact steps above and your hybrid doc's flow.