Supplementary Material

A. Atlas Generation

Several of our feature extraction strategies require a healthy reference atlas. For each dataset, we generate one by selecting 200 healthy subjects from the training set (KL grade 0 for OAI; cognitively normal for ADNI) and computing a population template using the normalization algorithm of [4] with *uniGradICON* as the registration backbone. These subjects are excluded from the validation and test splits to avoid data leakage. The resulting atlas serves as a fixed reference for registration-based feature extraction (atlas-s_t strategy) and for standardizing the alignment of volumes across subjects.

B. Affine Registration via Inverse Consistency by Construction

It is often desirable to affinely register images to a template before nonparametric registration, because the affine component of the transform might be a nuisance variable, e.g., due to inconsistent patient positioning. In addition, for large-scale analysis (in our experiments over 20,000 patient–template registrations) fast affine registration is essential. Prior work [33] reported that conventional approaches perform poorly for cross-subject affine registration on the OAI data. We therefore develop a deep-learning approach building upon inverse consistent by construction affine layers [14] trained on the uniGradICON [37] model. The architecture proposed in [14] performs affine and nonparametric registration via the architecture

$$\texttt{TSC}\{\Psi_1, \texttt{TSC}\{\Psi_2, \texttt{TSC}\{\Xi_1, \Xi_2\}\}\}\,, \tag{7}$$

where Ψ_i are inverse consistent affine networks, Ξ_i are inverse consistent by construction nonparametric networks, and TSC denotes the inverse consistent composition operator defined in [14]. To create an architecture that purely performs affine registration, we simply only use inverse consistent affine layers:

$$\texttt{TSC}\{\Psi_1, \texttt{TSC}\{\Psi_2, \texttt{TSC}\{\Psi_3, \texttt{TSC}\{\Psi_4, \Psi_5\}\}\}\}\,. \tag{8}$$

We train this affine-only model for 180 epochs at a low resolution of $88 \times 88 \times 88$, and upsample the resulting transforms to the original image resolution. The resulting registrations are accurate, achieving an average Dice score of 40% on femoral and tibial cartilage, compared to 38% for NiftyReg [33], while benefiting from the fast inference speed of deep learning models over optimization-based methods.

C. Implementation Details

All experiments are implemented in PyTorch [30] and run on NVIDIA RTX A6000 GPUs (49 GB VRAM). For each linear probe, we use full-batch AdamW [24] with learning rate 10^{-3}, train for 10,000 iterations, and select the best model based on validation accuracy (checkpointed every 1,000 iterations). We use class-weighted cross-entropy loss to address class imbalance. All experiments are repeated with 3 random seeds (42, 123, 456) and we report the mean across seeds.

For OAI, we use intersection splits such that all models are evaluated on the same 6,115 test samples across tables (the intersection of available, non-NaN features across all five models). For registration features, we concatenate the bottleneck features from the four U-Net sub-networks of uniGradICON/GradICON, yielding a 239K-dimensional feature vector per scan (or scan pair). Segmentation features are of dimension 197K (SAM-Med3D), 332K (SwinUNETR), and 92K (task-specific U-Net).

For ADNI, we use 70/15/15 subject-level stratified splits, excluding 200 CN subjects at screening that were used for atlas construction.

D. KLG Classification: Atlas-Difference Variants

For comparison with the main Table 2, we also evaluate atlas-difference features for the segmentation models, in which we subtract the atlas's segmentation feature from the scan's feature before classification. Atlas subtraction provides no significant improvement over the raw segmentation features.

Table 8. Atlas-difference variants of the segmentation models on OAI KLG classification (Acc, AUC %).

A	D	C	SAM-Med3D	SwinUNETR	UNet
			57.9, 79.7	54.4, 75.0	40.2, 57.9
✓			62.9, 84.5	58.7, 79.3	43.1, 63.5
✓		✓	66.3, 86.4	61.2, 82.0	42.8, 62.4
	✓		62.6, 83.5	62.2, 80.5	47.1, 69.3
	✓	✓	54.5, 76.7	56.1, 76.7	46.5, 59.8

E. Use of Large Language Models

During the preparation of this work, the authors used Claude and ChatGPT models to assist in refining the writing and reviewing experimental code. The authors take full responsibility for the content of the published manuscript.