Appendices for ContIG: Self-supervised Multimodal Contrastive Learning for Medical Imaging with Genetics

Aiham Taleb[1*] Matthias Kirchler[1,2*] Remo Monti[1] Christoph Lippert[1,3]

[1] Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany

[2] TU Kaiserslautern, Germany

[3] Hasso Plattner Institute for Digital Health at the Icahn School of Medicine at Mount Sinai, NYC, USA

{firstname.lastname}@hpi.de

A. Training & Implementation Details

A.1. Datasets Preprocessing

A.1.1 UK Biobank Genetic Modalities

During the pretraining phase using UK Biobank data, we choose the following feature dimensions. For the raw-SNPs, we uniformly sample every 100^{th} SNP from 22 Chromosomes (excluding the X and Y chromosomes), resulting in $7,854$ SNPs per sample. For PGS, we used 481 scores for a wide variety of different traits downloaded from the PGS Catalog [13]. We created burden scores for $18,574$ protein-coding genes [19]. These binary scores indicate whether a participant has at least one potentially damaging rare (MAF $< 1\%$) variant within a given gene.

A.1.2 Diabetic Retinopathy detection (APTOS)

In this task we use the APTOS 2019 Blindness Detection [1] dataset, which has $3,662$ retinal fundus training samples. As explained in the main paper, the labels in this dataset have five levels of disease severity, defining five classes. However, these classes are not mutually exclusive, as a higher disease severity of *e.g.* four is also of level three and below. Hence, we employ a multi-hot encoding scheme for the labels. For instance, class three is encoded as `[1,1,1,0,0]` and two as `[1,1,0,0,0]`, and so on. We split the dataset into three different splits of training (60%), validation (20%), and test (20%). There is no overlap of patients across these splits.

A.1.3 Retinal Fundus Disease Classification (RFMiD)

For this task, we use the Retinal Fundus Multi-disease Image Dataset (RFMiD) [20], which has $3,200$ images. The overall number of disease classes is 45. However, we found that two classes ("HR" and "ODPM") have no positive cases, so we exclude these two classes and only work with the remaining 43 classes. As mentioned before, we convert these classes to multi-hot labels and solve the task as multilabel classification. We use this dataset's official splits for training, validation, and test.

A.1.4 Pathological Myopia Segmentation (PALM)

We use the Pathologic Myopia challenge dataset [7] for this task, which has 400 image samples with segmentation masks. As for segmentation labels, this dataset has three annotated areas: i) peripapillary atrophy (available for 311 cases), ii) optic disc (available for all cases), and iii) detachment (available for 12 cases only). Given that detachment is rarely available, we omit it from this task and only predict the atrophy and disc classes. We stratify the patients using the atrophy labels, to ensure equal representation of classes in train (60% of dataset size) / val (20%) / test (20%) splits.

A.1.5 Cardiovascular Risk Prediction (UKB)

To predict the cardiovascular risk factors of (sex, age, BMI, SBP, DBP, smoking status) from retinal fundus scans, we use $102,219$ images from the UKB [26]. This corresponds to the training split (70% of UKB dataset size). We use the remaining scans for validation (10% of dataset size) and for the test split (20%). Each person only appears in one split. The training for this task is performed using two models: i) one model to classify the categorical labels (sex to binary labels $\{0,1\}$, smoking status to binary labels too), ii) a second model to predict – solved as a regression task – the remaining continuous variables (age, BMI, SBP, and DBP). We use two models because the loss values of these two tasks have different scales. We preprocess the values of the continuous

*Equal contribution

factors by standardization (removing the mean and scaling to unit variance). Finally, we impute the missing values of these factors by using the "mean" for continuous factors and "median" for discrete factors.

A.2. Imaging Preprocessing

A.2.1 Image Quality Control

The UK Biobank contains a relatively large number of retinal fundus images with bad quality (*e.g.* completely black or extremely overexposed). To filter out extreme outliers, we performed two steps of quality control. First, we only included images where a simple circle-detection algorithm [10] could find a circle. In the second step, we filtered out the top and bottom 0.5% brightest and darkest remaining images.

A.2.2 Image transformations

We cropped images to the circles detected in Appendix A.2.1 and rescaled to 448×448 pixels. During training, we randomly transform images by a rotation of up to 20° and flip the image horizontally with a 50% probability. We also follow the common practice of normalizing (standardizing) all the image intensities using the mean and standard deviation from ImageNet [5].

A.3. Genetics Preprocessing

In all our experiments we used the genetic data provided by the UK Biobank. The three different genetic modalities require different preprocessing steps, which we detail in this section.

A.3.1 Raw SNPs

The raw SNPs are a cross section of all SNPs collected on microarray chips, collecting approximately 800k genetic variants in total across all chromosomes. More information on data collection can be found at https://biobank.ctsu.ox.ac.uk/crystal/label.cgi?id=263.

The individual SNPs are coded in additive format, *i.e.* 0 stands for no deviation from the reference genome, 1 means that one of the two chromosome copies has a deviation and the other not, and 2 means that both chromosome copies show a deviation from the reference genome. We treated SNPs as continuous variables (opposed to, *e.g.* separating them into three classes each) and imputed missing values by mode imputation. Since 800k feature dimensions are challenging to handle, and SNPs are highly spatially correlated along the genome [22], we only sampled every 100-th SNP from the full microarray. We also only included SNPs on the 22 autosomal (=not sex-specific) chromosomes, as handling sex chromosomes requires special statistical care and leads to non-shared features between genetic males and females. Together, this means we include 7,854 SNPs in our models.

A.3.2 Polygenic Risk Scores

For computing polygenic risk scores, we downloaded all PGS weight files included in the PGS Catalog [13] (https://ftp.ebi.ac.uk/pub/databases/spot/pgs/, last accessed October 11, 2021; at the time of writing, a large batch of new scores has been added to the PGS catalog), a collection of published PGS. The PGS files provide weights for a linear model to compute risk scores from the raw genetic data. To have a large intersection of available SNPs for our UKB population and the weights provided by the PGS catalog, instead of using the raw microarray data from Appendix A.3.1, we used *imputed* data. The imputed data uses prior knowledge about correlations between SNPs collected and not collected on the respective microarray ("linkage disequilibrium", LD) to infer the missing features with high accuracy. Imputed data was precomputed by the UKB, and more information can be found at https://biobank.ctsu.ox.ac.uk/crystal/label.cgi?id=100319. Using the imputed data, we computed 481 polygenic scores for our cohort using the PLINK software [21], ignoring scores that gave errors or that only recorded genome positions in a different reference genome build.

For some traits, there are multiple distinct risk scores in the PGS catalog, as multiple independent studies have been performed on the same trait. For example, the trait "melanoma" appears 9 times in our subset of selected PGS scores, while other traits, such as "insomnia" appear only once. The scores contain partially overlapping genetic markers, and the number of SNPs used for each individual score vary from only 1 to several millions.

A.3.3 Burden Scores

We ran the Functional Annotation and Association Testing Pipeline [19] to functionally annotate all the genetic variants present in the UK Biobank 200k exome sequencing release [27]. Protein loss of function and missense variants that were predicted to be damaging were used to construct burden scores across all protein coding genes. We considered only rare variants with minor allele frequencies below 1%. Of these variants 41% were "singletons", i.e. only observed once in our sample. Specifically, each participant was assigned a binary vector of length 18,574 corresponding to the number of protein coding genes. For every gene, the entry in this vector is 1 if the participant harbored at least one potentially damaging variant in that gene, or 0 if no potentially damaging variants were observed in that gene for that participant. This coding has been applied in

rare-variant association studies in order to aggregate the effects of many rare variants within genes, where it can boost statistical power and reduce the burden of multiple testing [14, 19].

A.4. Training Details

We provide the training details for all pretraining (self-supervised) and downstream tasks in this section.

- **Batch sizes**: we use a unified batch size of 64 across all pretraining and downstream tasks.
- **Optimizers**: we use Adam optimizer [11] in all pretraining and downstream tasks.
- **Schedulers**: during self-supervised pretraining (with ContIG and the baselines), we decay the learning rate with the cosine decay schedule without restarts [18].
- **Learning rates**: we use an initial learning rate of 0.001 across all tasks. However, we reduce the learning rate during training in the PALM semantic segmentation task to 1×10^{-4} after 10 warum epochs.
- **Weight decay**: in pretraining tasks, we use a weight decay factor of 1×10^{-6}. In downstream tasks, we use a weight decay factor of 1×10^{-5}.
- **Number of epochs**: in pretraining tasks, we train all models for 100 epochs. In downstream tasks, we finetune for:
  - For the PALM, APTOS, and RFMiD tasks: we train all models for 50 epochs.
  - For Cardiovascular risk prediction tasks: we finetune all models for 5 epochs (≈ 8000 steps).
- **Network architectures**: for the *image encoder*, as mentioned before, we use a Resnet50 [9] architecture across all pretraining and downstream tasks. For the *genetics encoders*, we vary between following choices:
  - None: here we do not have any hidden fully-connected layer for the genetics, and we feed them as inputs to the projection head directly.
  - H1: we process the genetic inputs with one hidden layer of size 2048. (followed by a `ReLU` activation and `Batchnorm1D` layers)
  - H12: we process the genetics with two hidden layers, both of size 2048. (Each layer is followed by a `ReLU` and `Batchnorm1D`)

  For the *projection head*, we follow [3] in using two fully-connected layers. The first has a size of 2048 and is followed by a `ReLU`. The second has size of 128, which is the projection embedding size. Finally, for classification and regression downstream tasks we add one fully-connected `Linear` layer on top to perform the task. But for the *PALM segmentation* task, we add a U-Net [23] decoder on top of the Resnet50 encoder. For upsampling layers in the decoder, we use transposed convolutional layers `ConvTranspose2d`.
- **Loss functions**: the used loss functions for each task are as follows:
  - ContIG: for training our method, we use a contrastive loss (`NTXentLoss`). This loss is implemented using a cross-entropy loss, where the model is trained to classify which sample is positive in each mini-batch. However, our version of the `NTXentLoss` only does inter-modal contrasting, and not intra-modal. We set $\lambda = 0.75$ in this loss (Eq. 1 in the main paper), and the temperature $\tau = 0.1$. Note that a larger value of λ gives more importance to image features than genetic features.
  - APTOS & RFMiD: we use the binary cross-entropy loss in both tasks.
  - PALM: we use a weighted combined loss of Dice-loss [25] (weight=0.8) and binary cross-entropy (weight=0.2).
  - Cardiovascular risk classification (sex & smoking status): we use a binary cross-entropy loss.
  - Cardiovascular risk prediction (age & BMI & SBP & DBP): we use the Mean Square Error (MSE) loss.
  - SimCLR [3]: this method uses the contrastive `NTXentLoss` too. We similarly set the temperature $\tau = 0.1$.
  - NNCLR [6]: this method uses the contrastive `NTXentLoss` too. We similarly set the temperature $\tau = 0.1$.
  - Simsiam [4]: this method does not use negative sampling, and instead uses a Siamese network to minimize the similarity between two augmented views of the same image. Hence, the loss function used is the negative cosine similarity loss.
  - BYOL [8]: this method has the same loss used in Simsiam, which is the negative cosine similarity.
  - Barlow Twins [29]: this method modifies the contrastive loss to compute the cross-correlation matrix between two sets of embeddings, which are for the same batch of images but with different image augmentations. Then, it tries to make this matrix close to the identity matrix.

A.5. Implementation Details

We implement all of our methods using `Python`. The libraries we rely on are `PyTorch v1.9.1`, `Pytorch-Lightning v1.4.8`, `torchvision v0.10.0`, `torchmetrics v0.4.0`, and `Lightly` [15] (for baseline self-supervised implementations). We also follow the reproducibility instructions for `Pytorch-Lightning` [16], *i.e.* by setting a unified random seed of 42 for all scripts and workers, and by using `deterministic` algorithms. We attach our source code

with this supplementary material submission.

B. Additional Downstream Results

B.1. Complete Finetuning Results

In this section, we present the full set of results for finetuning our ContIG models versus the same baselines. These extended evaluation results, in Tab. 1, show that ContIG is advantageous to the baselines. The rows in Tab. 1 are grouped in the following order: i) baseline trained from scratch, ii) self-supervised baselines, iii) ContIG trained on single genetic modalities with the images, and iv) ContIG trained on multiple genetic modalities with images.

B.2. Linear Evaluation Results

In this section, we follow a linear evaluation protocol [3, 28, 30], meaning that the encoder weights are kept frozen and we only train a linear classifier / regressor on top. As shown in Tab. 2, models trained with our method "ContIG" consistently outperform the baselines. Linear evaluation aims to provide a good idea about the quality of semantic representations stored in the model encoder.

B.3. Data-Efficiency Results

In this section, we assess the quality of semantic representations in a semi-supervised experimental scheme. We choose randomly 1% and 10% of the labels provided by UK Biobank (UKB) [26], and perform the downstream tasks of Cardiovascular Risk Factors prediction. Then, we evaluate using the same fixed test split of 20% of UKB dataset size. We choose this particular downstream task as UKB's dataset size is large enough to allow a simulation for expert annotation collection process, *i.e.* 1% of number of overall labels is approximately 1000 samples, and such number may simulate an annotation process. The other benchmark datasets (APTOS [1], RFMiD [20], and PALM [7]) are relatively small in size. The evaluation results shown in Tab. 3 compare models trained with ContIG to models trained with the self-supervised baselines. ContIG outperforms the baselines in this evaluation scheme too. Note that all models are trained on the same exact subset of individuals and also evaluated on the same test set. The results for this data-efficient evaluation scheme especially confirm the advantages of pretraining with multiple genetic modalities using the "Outer" aggregation scheme. Notably, semi-supervised pretraining of ContIG with only 1% labeled data still outperforms the self-supervised baselines when they have $10\times$ as much labeled data available.

C. Additional Feature Explanation Results

C.1. Method Validation

We ran a baseline experiment to validate that our feature explanation method properly attributes to meaningful features. In this experiment, instead of genetic features, we use phenotypic covariates such as age, sex, systolic and diastolic blood pressure (SBP and DBP), which can be predicted reliably from retinal fundus images. Additionally, we include the first 40 principal components, which mostly capture population structure information. As control variables, we also feed five random noise variables into the training process, which have no association with the images at all. Fig. 1 shows the aggregated feature explanations. As expected, the noise variables (`noise0, ... noise4`) get assigned very low explanation scores, while all other variables have considerable influence. This validates that our feature explanation approach can distinguish between variables that carry true information relevant to the network and variables that are unrelated to the images.

C.2. Multimodal Explanation Results

Fig. 2 shows the aggregated attribution scores for each of the three modalities, Raw-SNPs, PGS, and Burdens, for ContIG with the "Outer" training scheme. Fig. 2a shows that PGS scores on average have more influence than individual SNPs or burden scores. However, Fig. 2b also shows that that in aggregate, raw SNPs and burden scores have more total influence on the model. This is likely due to PGS only having 481 features, while raw SNPs and Burdens have 7,854 and 18,574 features, respectively. This may also explain the small but counterintuitive performance drop from ContIG (PGS) to ContIG (Outer RPB): the strongest signal, PGS, gets "drowned out" by the less important but overabundant signal in the raw SNPs and burden scores.

D. Ablation Study

We conducted ablations for the hyper-parameters of training batch size (b) and lambda (λ) –from Eq. 1, used in the pretraining phase using our method ContIG. For the batch size, due to memory limits of available GPUs, 64 multimodal samples is the maximum we could fit. We consider drawing negative samples from a memory bank instead a future work. Despite that, our method already outperforms SOTA in downstream tasks (see Tab. 1 and Tab. 2). Therefore, we try smaller batch sizes of $32, 16$, and as expected, we observe a slight drop in downstream performance ($\leq 2\%$). For lambda, we evaluate the values of $0.5, 0.25$, and also obtain comparable results ($\leq 1\%$). These results show that our method exhibits an improved robustness to smaller batch sizes and lambda values.

Model & Genetics Encoder		APTOS	RFMiD	PALM	Cardio. Risk Pred.	
		QwKappa ↑	ROC-AUC ↑	Dice-Score ↑	MSE ↓	ROC-AUC ↑
Baseline	-	80.47	91.64	77.25	3.440	56.29
SimCLR [3]	-	81.83	91.88	70.41	3.451	59.38
SimSiam [4]	-	75.44	91.28	72.26	3.442	57.37
BYOL [8]	-	71.09	89.88	66.32	3.414	59.73
Barlow Twins [29]	-	72.28	92.03	70.53	3.430	59.05
NNCLR [6]	-	77.93	91.89	72.06	3.426	61.95
ContIG (Raw-SNP)	None	81.99	92.27	74.96	3.366	64.71
ContIG (Raw-SNP)	H1	84.01	93.22	76.98	3.254	70.10
ContIG (Raw-SNP)	H12	82.56	93.09	77.02	3.201	69.58
ContIG (PGS)	None	83.84	91.63	76.86	3.257	69.81
ContIG (PGS)	H1	<u>85.93</u>	93.31	**78.47**	<u>3.176</u>	**72.72**
ContIG (PGS)	H12	**86.44**	93.04	77.04	3.216	70.69
ContIG (Burden)	None	82.92	**93.68**	76.89	3.273	71.91
ContIG (Burden)	H1	83.22	93.03	76.49	**3.160**	<u>72.37</u>
ContIG (Burden)	H12	83.61	93.14	76.72	3.236	71.50
ContIG (Inner RPB)	None	83.49	93.31	77.11	3.195	71.68
ContIG (Inner RPB)	H1	81.52	92.95	77.34	3.202	70.80
ContIG (Inner RPB)	H12	80.24	92.94	75.37	3.235	68.89
ContIG (Outer RPB)	None	82.93	93.01	76.31	3.260	69.16
ContIG (Outer RPB)	H1	84.22	<u>93.62</u>	76.97	3.187	71.80
ContIG (Outer RPB)	H12	84.21	93.41	<u>77.51</u>	3.233	71.13

Table 1. Downstream evaluation results by fine-tuning on each task. **Bold** indicates the best result, <u>underlined</u> is second best. RPB in our method stand for the genetic modalities used: *R*aw-SNPs, *P*GS-scores, and *B*urden-scores. ↑ means higher is better, and ↓ lower is better.

Model & Genetics Encoder		APTOS	RFMiD	PALM	Cardio. Risk Pred.	
		QwKappa ↑	ROC-AUC ↑	Dice-Score ↑	MSE ↓	ROC-AUC ↑
SimCLR [3]	-	35.02	86.53	59.77	3.998	52.26
SimSiam [4]	-	21.25	87.91	56.58	3.998	53.13
BYOL [8]	-	17.39	87.84	54.04	4.009	52.29
Barlow Twins [29]	-	44.75	87.65	59.52	3.952	54.28
NNCLR [6]	-	24.76	85.80	66.25	3.870	54.17
ContIG (Raw-SNP)	None	59.14	89.24	72.82	3.683	59.07
ContIG (Raw-SNP)	H1	69.85	89.99	75.25	3.443	64.36
ContIG (Raw-SNP)	H12	68.72	90.47	74.39	3.439	69.58
ContIG (PGS)	None	66.34	88.16	75.03	3.488	62.64
ContIG (PGS)	H1	**72.38**	90.43	76.35	3.426	63.98
ContIG (PGS)	H12	70.20	90.01	**77.13**	3.481	63.27
ContIG (Burden)	None	70.29	<u>91.08</u>	75.31	3.453	64.72
ContIG (Burden)	H1	70.67	90.62	75.42	3.421	64.70
ContIG (Burden)	H12	<u>71.22</u>	**91.10**	76.09	3.434	<u>64.84</u>
ContIG (Inner RPB)	None	70.26	89.94	75.27	3.439	63.84
ContIG (Inner RPB)	H1	66.94	88.65	75.00	3.404	64.73
ContIG (Inner RPB)	H12	68.41	90.56	73.08	3.457	63.45
ContIG (Outer RPB)	None	66.94	90.38	75.29	3.448	**65.20**
ContIG (Outer RPB)	H1	66.60	89.46	<u>77.04</u>	<u>3.398</u>	64.59
ContIG (Outer RPB)	H12	68.57	90.51	76.50	**3.388**	**65.20**

Table 2. Downstream evaluation results by linear evaluation on each task. Similarly, the results obtained by ContIG outperform all baselines. **Bold** indicates the best result, <u>underlined</u> is second best. RPB in our method stand for the genetic modalities used: *R*aw-SNPs, *P*GS-scores, and *B*urden-scores. ↑ means higher is better, and ↓ lower is better.

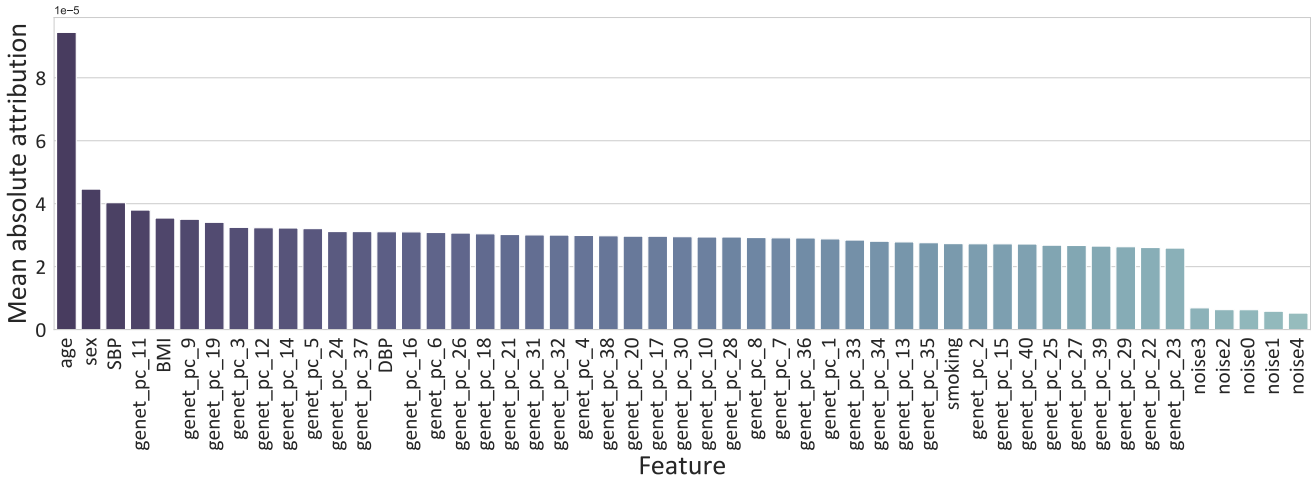
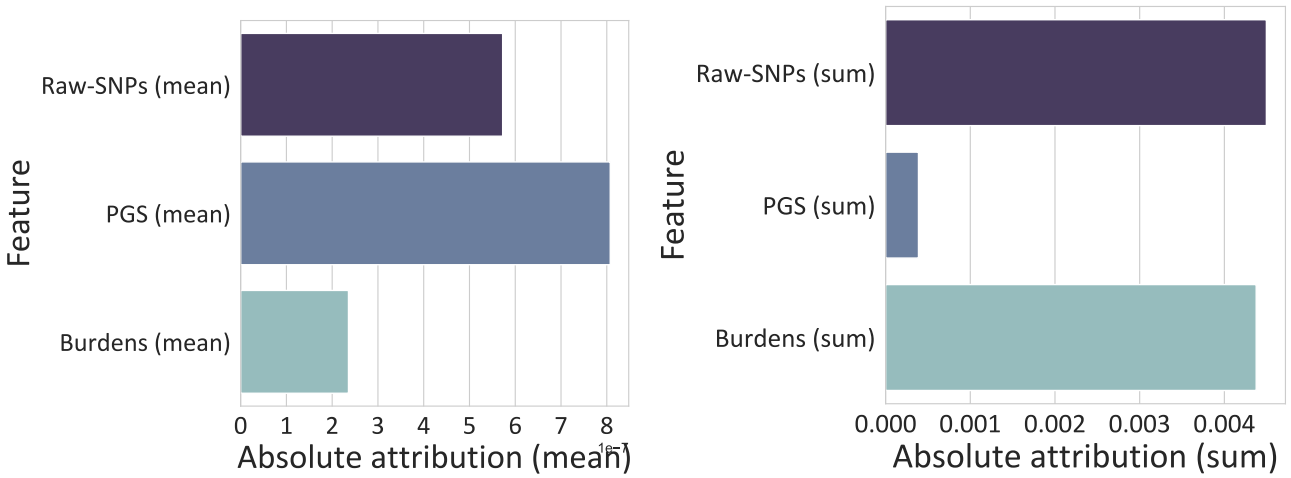


Figure 1. Explanation method validation. Shown is the mean absolute attribution for each feature aggregated over a batch-size of 1,000 individuals. `noise0, ..., noise4` don't carry any information and also get downweighted by our attribution method.



(a) Absolute attribution for each modality, aggregated by mean.

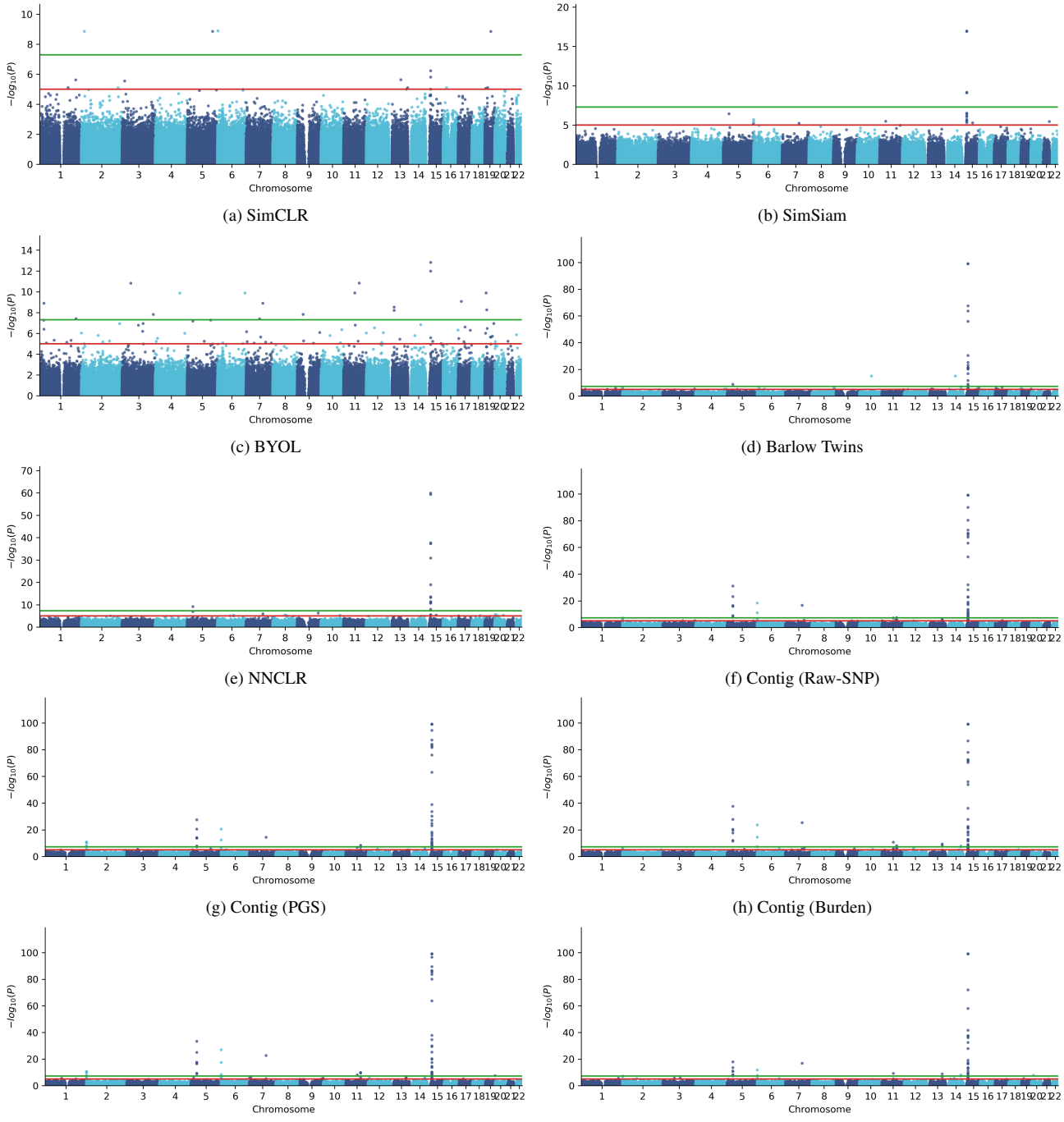
(b) Absolute attribution for each modality, aggregated by sum.

Figure 2. Absolute attributions by modality for ContIG (Outer RPB).

E. GWAS Analysis Details

We produced feature vectors by computing the hidden-layer embedding for each image in the test-split of our dataset (10% of the whole dataset, 7,079 individuals). In contrast to the main training, we only used embeddings of the left eye and only included each individual once. Feature vectors were reduced to 10 dimensions using a PCA. Before computing the association results, we also used an inverse-normal transform [24] after conditioning on the potential confounders "sex", "age", as well as the first 15 genetic PCs. This ensures that the residuals of the marginal distributions are approximately normally distributed and outlier deviations from normality don't artificially inflate the type-1 error rate, leading to spurious correlations. We performed the genetic association study with the PLINK2 software [2], using a linear model for each of the ten dimensions individually. We again correct for the same confounders in the linear model. Finally, we aggregate the summary statistics of the ten individual features into a single p-value for each SNP by using a Bonferroni-correction of the factor 10, following [12].

Genetic variants are locally highly correlated. Therefore, we group significantly associated SNPs that are spatially close and in LD together using the PLINK [21] clumping functionality (using parameters `clump-p1` $= 5 \cdot 10^{-8}$, `clump-p2` $= 10^{-7}$, `clump-r2` $= 0.1$, `clump-kb` $=$ 150). We reported the number of independent associated regions returned by this procedure in the main document.



(a) SimCLR (b) SimSiam

(c) BYOL (d) Barlow Twins

(e) NNCLR (f) Contig (Raw-SNP)

(g) Contig (PGS) (h) Contig (Burden)

(i) Contig (Inner RPB) (j) Contig (Outer RPB)

Figure 3. Manhattan plot for the GWAS with different training methods. The x-axis shows the position of each SNP on the genome, the y-axis is the negative base-10 logarithm of the p-value for each SNP. Higher values correspond to lower p-values, correspond to stronger signal. The red line corresponds to a significance threshold of 0.05 Bonferroni-adjusted for the number of SNPs; the green line corresponds to "genome-wide significance" ($5 \cdot 10^{-8}$). P-values are clamped at 10^{-99} for clearer visualization (only relevant for the loci on chromosome 15 with a minimum p-value of 10^{-320}). Note the different y-axis scales.

Fig. 3 shows the manhattan plot of genome-wide associations from the GWAS with ContIG and other pretraining methods. A number of very strong signals, e.g. on chromosomes 15 and 5, are known to be associated with skin

Model	Label Fraction			
	1%		10%	
	MSE ↓	ROC ↑	MSE ↓	ROC ↑
SimCLR [3]	4.029	51.43	3.762	54.29
SimSiam [4]	3.861	53.35	3.564	57.45
BYOL [8]	3.894	51.68	3.505	56.71
Barlow Twins [29]	3.788	51.89	3.558	56.86
NNCLR [6]	3.913	52.20	3.643	55.99
ContIG (Raw-SNP)	3.541	<u>60.11</u>	3.414	64.81
ContIG (PGS)	3.521	59.23	<u>3.391</u>	<u>65.86</u>
ContIG (Burden)	3.540	59.74	3.393	65.41
ContIG (Inner RPB)	<u>3.511</u>	59.95	3.397	65.71
ContIG (Outer RPB)	**3.490**	**60.39**	**3.378**	**65.99**

Table 3. Data-efficient evaluation results by fine-tuning on subsets of UKB samples. All our ContIG models use the "H1" genetic encoder variant. **Bold** indicates the best result, <u>underlined</u> is second best. ↑ means higher is better, and ↓ lower is better.

b/λ	APT	RFM	PLM	UKB	
	QwK ↑	roc ↑	Dice ↑	MSE ↓	roc ↑
64/0.75	**86.33**	**93.92**	**77.56**	3.180	72.65
64/0.5	84.13	93.52	77.32	**3.167**	**73.08**
64/0.25	84.91	93.77	76.64	3.174	72.37
32/0.75	84.01	93.41	76.59	3.182	72.11
16/0.75	84.09	92.77	76.40	3.296	67.41

Table 4. Ablation results for batch-size (b) and lambda (λ).

pigmentation and cardiovascular traits. Manhattan plots for the other pretrained models look similar but with less signal. Almost all models found the very strong signals on chromosome 15. Interestingly, the manhattan plots for both SimCLR and BYOL (Figures 3a & 3c) show clear signs of a ill-fitted association model, with many (for BYOL) but small, most likely spurious associations distributed over the whole genome but no signal in the chromosome-15 pigmentation region. This happens even after applying the inverse-normal transformation to counteract outliers and is likely due to different forms of confounding. This finding also explains the surprisingly large number of hits for BYOL – they are most likely false-positives. A more careful analysis with mixed effect models [17] and in-depth inspection of the image features is beyond the scope of this article.

References

[1] APTOS. Aptos 2019 blindness detection. https://www.kaggle.com/c/aptos2019-blindness-detection/. Accessed: 2021-11-04. 1, 4

[2] Christopher C Chang, Carson C Chow, Laurent CAM Tellier, Shashaank Vattikuti, Shaun M Purcell, and James J Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience*, 4(1):s13742–015, 2015. 6

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Int. Conf. on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. 3, 4, 5, 8

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 3, 5, 8

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, Miami, FL, USA, 2009. IEEE. 2

[6] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, pages 9588–9597, October 2021. 3, 5, 8

[7] Huazhu Fu, Fei Li, José Ignacio Orlando, Hrvoje Bogunović, Xu Sun, Jingan Liao, Yanwu Xu, Shaochong Zhang, and Xiulan Zhang. Palm: Pathologic myopia challenge, 2019. 1, 4

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 3, 5, 8

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[10] John Illingworth and Josef Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):690–698, 1987. 2

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 3

[12] Matthias Kirchler, Stefan Konigorski, Matthias Norden, Christian Meltendorf, Marius Kloft, Claudia Schurmann, and Christoph Lippert. transfergwas: Gwas of images using deep transfer learning. *bioRxiv*, 2021. 6

[13] Samuel A Lambert, Laurent Gil, Simon Jupp, Scott C Ritchie, Yu Xu, Annalisa Buniello, Aoife McMahon, Gad Abraham, Michael Chapman, Helen Parkinson, et al. The polygenic score catalog as an open database for reproducibility and systematic evaluation. *Nature Genetics*, 53(4):420–425, 2021. 1, 2

[14] Seunggeun Lee, Mary J Emond, Michael J Bamshad, Kathleen C Barnes, Mark J Rieder, Deborah A Nickerson, ESP Lung Project Team, David C Christiani, Mark M Wurfel, Xihong Lin, et al. Optimal unified approach for rare-variant association testing with application to small-sample case-control whole-exome sequencing studies. *The American Journal of Human Genetics*, 91(2):224–237, 2012. 3

[15] lightly.ai. lightly. https://github.com/lightly-ai/lightly. Accessed: 2021-11-20. 3

[16] PyTorch Lightning. Reproducibility. https://pytorch-lightning.readthedocs.io/en/latest/common/trainer.html#reproducibility. Accessed: 2021-11-19. 3

[17] Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature methods*, 8(10):833–835, 2011. 8

[18] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. 3

[19] Remo Monti, Pia Rautenstrauch, Mahsa L Ghanbari, Alva Rani James, Uwe Ohler, Stefan Konigorski, and Christoph Lippert. Identifying interpretable gene-biomarker associations with functionally informed kernel-based tests in 190,000 exomes. *bioRxiv*, 2021. 1, 2, 3

[20] Samiksha Pachade, Prasanna Porwal, Dhanshree Thulkar, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Luca Giancardo, Gwenolé Quellec, and Fabrice Mériaudeau. Retinal fundus multi-disease image dataset (rfmid): A dataset for multi-disease detection research. *Data*, 6(2), 2021. 1, 4

[21] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American journal of human genetics*, 81(3):559–575, 2007. 2, 6

[22] David E Reich, Michele Cargill, Stacey Bolk, James Ireland, Pardis C Sabeti, Daniel J Richter, Thomas Lavery, Rose Kouyoumjian, Shelli F Farhadian, Ryk Ward, et al. Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204, 2001. 2

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *MICCAI*, pages 234–241. Springer International Publishing, 2015. 3

[24] Tamar Sofer, Xiuwen Zheng, Stephanie M Gogarten, Cecelia A Laurie, Kelsey Grinde, John R Shaffer, Dmitry Shungin, Jeffrey R O'Connell, Ramon A Durazo-Arvizo, Laura Raffield, et al. A fully adjusted two-stage procedure for rank-normalization in genetic association studies. *Genetic epidemiology*, 43(3):263–275, 2019. 6

[25] Sorensen-Dice. Dice score. https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient. Accessed: 2021-11-05. 3

[26] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott,

Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory Collins. Uk biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLOS Medicine*, 12(3):1–10, 03 2015. 1, 4

[27] Joseph D Szustakowski, Suganthi Balasubramanian, Erika Kvikstad, Shareef Khalid, Paola G Bronson, Ariella Sasson, Emily Wong, Daren Liu, J Wade Davis, Carolina Haefliger, et al. Advancing human genetics research and drug discovery through exome sequencing of the uk biobank. *Nature genetics*, 53(7):942–948, 2021. 2

[28] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 4

[29] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 3, 5, 8

[30] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 649–666. Springer International Publishing, 2016. 4