Flexible Models of Functional Annotations to Variant Effects using Accelerated Linear Algebra

Alan Nawzad Amin*
New York University
alanamin@nyu.edu

Andres Potapczynski*
New York University
andpotap@nyu.edu

Andrew Gordon Wilson
New York University
aw130@nyu.edu

Abstract

To predict and understand the causes of disease, geneticists build models that predict how a genetic variant impacts phenotype from genomic features. There is a vast amount of data available from the large projects that have sequence hundreds of thousands of genomes; yet, state-of-the-art models, like LD score regression, cannot leverage this data as they lack flexibility due to their simplifying assumptions. These models use simplifying assumptions to avoid solving the large linear algebra problems introduced by the genomic correlation matrices. In this paper, we leverage modern fast linear algebra techniques to develop WASP (genome Wide Association Studies with Preconditioned iteration), a method to train large and flexible neural network models. On semi-synthetic and real data we show that WASP better predicts phenotype and better recovers its functional causes compared to LD score regression. Finally, we show that training larger WASP models on larger data leads to better explanations of phenotypes.

1 Introduction

To predict the risk of genetic disease and understand its molecular causes, Genome Wide Association Studies (GWAS) use data from up to hundreds of thousands of individuals to build models that correlate the presence of genetic variants with phenotypes such as disease or height (Yang et al., 2010; Visscher et al., 2017; Halldorsson et al., 2021). However there are orders of magnitude more variants than measurements, making GWAS underpowered to predict phenotype or determine the effects of all but the most impactful variants.

To increase prediction accuracy and uncover the molecular causes of disease, geneticists have leveraged the fact that complex phenotypes are extremely polygenic – that is, they are affected by a huge number of variants spread throughout the genome (Manolio et al., 2009; Boyle et al., 2017). Geneticists look for features that distinguish variants that do and do not effect a phenotype on a set of chromosomes and use these features to build "functionally informed" priors to analyze variants on other chromosomes (Gusev et al., 2014; Finucane et al., 2015; Kichaev et al., 2019). To build these priors they use functional genomic features (ENCODE Project Consortium, 2012; Lizio et al., 2015), such as measurements of DNA "open-ness" or binding of transcription factor proteins near the variant; and comparative genomics features (Cooper et al., 2005; Pollard et al., 2010), such as whether the variant is in a region of the genome that is conserved across primates. As more accurate measurements of genomic features are made and more individuals have their genomes sequenced, in principle, geneticists should be able to build more accurate functionally informed priors with more flexible models that learn from more features. In practice, however, significant computational challenges have prevented the development of large models.

Functionally informed priors are typically phrased as priors on the effect of each variant in a hierarchical Bayesian model of the genetic and phenotypic data (Loh et al., 2015; Zheng et al., 2024). Ideally, we could fit the prior using an empirical Bayes approach to maximize the marginal likelihood (Ni et al., 2018). Unfortunately this is numerically challenging due to linkage disequilibrium (LD) – the presence of variants in the genome can be strongly correlated, and accounting for this correlation in the marginal likelihood

*Equal contribution.

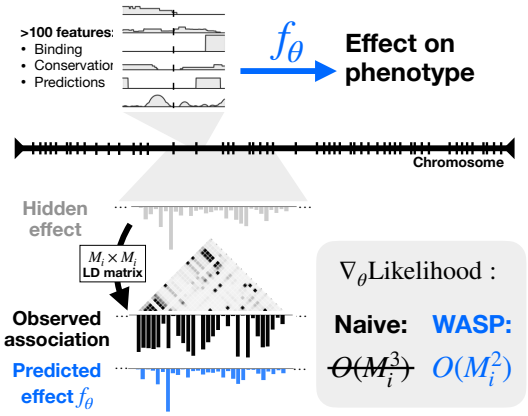

Figure 1: **WASP enables training large models to predict the effect of variants from genomic features by leveraging fast linear algebra.** Top: We want to train a model, f_θ, to predict the effect of a variant in our genome from a large set of curated genomic features in a window around the variant. Bottom: We train f_θ to maximize the likelihood of observed associations between variants and traits. We efficiently compute the likelihood by applying accelerated linear algebra on the correlation matrix of variants in a sliding window. See section 3 for full details.

involves inverting and calculating the log determinant of a large matrix known as the LD matrix. To avoid inverting this matrix, state-of-the-art methods sacrifice statistical efficiency by fitting summary statistics or an approximation of the marginal likelihood, or fit simple parametric models of the relation between functional annotations and phenotype (Li et al., 2024; Huang et al., 2024).

The challenge of having to invert a large matrix to perform empirical Bayesian inference was addressed in works on Gaussian process regression with two strategies (Gardner et al., 2018). First, using an iterative algorithm, inversion of an $M \times M$ matrix could be reduced from $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2K)$ where $K << M$ is the number of iterations; these algorithms have also been used for inverting large matrices in GWAS (Loh et al., 2015). Second, by building an approximate inverse to the large matrix which is easy to invert – a "preconditioner" – the number of steps K could be reduced by orders of magnitude.

Here we introduce a method to train large models that predict variant effects from functional annotations – genome Wide Association Studies with Preconditioned iteration (WASP) (Fig. 1). We outline our contributions:

- We amortize the cost of training large neural networks on phenotype association data with millions of variants by leveraging a banded approximation to the LD matrix and using the approximating slices as mini-batches during training.
- We introduce a specialized structured preconditioner that in conjunction with iterative algorithms allows us to efficiently perform challenging linear algebra operations at each training step.
- We show that training models with WASP leads to better fits to the data than the previous state-of-the-art LD score regression.
- We curate a large set of genomic features to train functional priors.
- We, for the first time, train large functionally informed priors on large public phenotype association data with WASP and explore the effect of model size and genomic features on the accuracy of the model.

Detailed relted work is in App. A. Our code for training WASP models is available at `https://anonymous.4open.science/r/fast_gen-05C2/`.

2 Background

In this section we describe models that describe traits using variants in the genome.

Functionally informed priors to predict variant effect To learn the heritibility of a trait, suppose that we have measured the genotypes of N ($\approx 10^5$) subjects – we have measured the presence or absence of variants at M ($\approx 10^6 - 10^8$) positions or alleles on both chromosomes – to get a genotype matrix $\tilde{X} \in \{0, 1, 2\}^{M \times N}$, and the presence of the trait to get a phenotype vector $y \in \mathbb{R}^N$. We can assume y is centered to have mean 0 and variance 1 and X is a centered $\tilde{X}$ with all rows mean 0 and variance 1.

Measured traits that we are interested in, such as height, smoking status or schizophrenia, are polygenic – they are influenced by many variants scattered throughout the genome rather than a small number of alleles (Manolio et al., 2009; Boyle et al., 2017). This is captured by the infinitesimal model in which each variant has a small effect drawn from a prior (Barton et al., 2016; Trippe et al., 2021).

A popular infinitesimal model is the linear model $y = X^\intercal \beta + \epsilon$ with iid noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ where the effect size at position m, β_m, is independently drawn from a prior normal distribution $\beta_m \sim \mathcal{N}(0, f_m)$. Therefore we can describe the marginal distribution of y as

$$y \sim \mathcal{N}\left(0, X^\intercal F X + \sigma^2 I\right), \text{ where } F = \mathrm{diag}(f). \tag{1}$$

Our first goal is estimating the effect size β. The challenge is that there are many more variables than observations , $N << M$, so it is challenging to get enough statistical power to predict the values of many β. Our second goal is to identify the features that characterize variants m with large effect sizes β_m.

We can achieve these goals with a good prior f, which will increase our power to determine β_m and predict which variants are expected to have large magnitude since $\mathbb{E}\beta_m^2 = f_m$. To build such a prior, we can take advantage of large datasets of genomic features C_m (elaborated in Sec. 4), to predict f_m with a model with parameters θ, $f_\theta(C_m)$. Naively, we may train f_θ and σ^2 by maximizing the marginal likelihood of Eqn. 1.

Summary Statistics However, to protect the privacy of study participants, we are given "summary statistics" rather than the precise value of y and X. In particular, we are given the empirical correlation matrix known as the "Linkage Disequilibrium (LD) matrix" $R = \frac{1}{N} X X^\intercal$; and the empirical associations $\hat{\beta} = \frac{1}{N} X y$. We can then write Eqn 1 in terms of summary statistics with $\sigma_N^2 \equiv \frac{1}{N}\sigma^2$:

$$\hat{\beta} \sim \mathcal{N}\left(0, RFR + \sigma_N^2 R\right). \tag{2}$$

The second term in the variance, $\sigma_N^2 R$, comes from spurious correlations with the noise ϵ; if the presence of two variants m and m' are correlated ($R_{m,m'}$ is large) then the associations $\hat{\beta}_m$ and $\hat{\beta}_{m'}$ will have similar correlations with the noise ϵ. The first term in the variance comes from the effect variants have on the trait. Specifically, the m, m' entry of RFR is $\sum_k R_{m,k} R_{m',k} f_k$, which is large if there are variants k correlated to both m and m' – large $R_{m,k}$ and $R_{m',k}$ – which are expected to have large effect f_k.

Now, in principle, we could build a prior by maximizing the likelihood of Eqn. 2:

$$-\frac{1}{2}\hat{\beta}^T \left(RF_\theta R + \sigma_N^2 R\right)^{-1} \hat{\beta} - \frac{1}{2}\log\left|RF_\theta R + \sigma_N^2 R\right| + c \tag{3}$$

where c is a constant value. The challenge is the need to calculate F_θ and then invert and calculate the log determinant of the huge $M \times M$ matrix R.

LD score regression (LDSR) Other methods have been devised for fitting θ while avoiding the explicit inversion. In this section we describe the most popular of these methods before moving to our method.

From Eqn 2 we can note that a variant m expected to have a large association if it is correlated to other variants expected to have large effects: [1]

$$\mathbb{E}[N\hat{\beta}_m^2] = N \sum_{m'} f_{m'} R_{mm'}^2 + \sigma^2. \tag{4}$$

The simplest model of heritability gives each variant the same expected heritability, $f_m = f$, in which case we can write Eqn. 4 as $\mathbb{E}[N\hat{\beta}_m^2] = Nf\left[\sum_{m'} R_{mm'}^2\right] + \sigma^2$. The term in the brackets, known as the LD score, measures how many other variants m is correlated with and can be precomputed before fitting f. By fitting a line to the magnitudes of the association statistics $N\hat{\beta}_m^2$ and precomputed LD scores, one can recover Nf as the slope and σ^2 as the intercept. This method, known as LD score regression, gives accurate predictions of how much of a trait is explained by genetics, f, and how much is caused by noise or the environment, σ^2 (Bulik-Sullivan et al., 2015). Finucane et al. (2015) extended this approach to fit a linear f_m that depends on d genomic features – in this case one performs a multi-dimensional linear regression with d precomputed variables.

More generally, one can in principle fit the linear relation $N\hat{\beta}^2$ against $NR^{\circ 2}F_\theta + \sigma^2\mathbb{1}$ for more flexible models f_θ. This method does not require inverting R; however, this method loses statistical efficiency by not making use of correlations between $\hat{\beta}$ (Ni et al., 2018).

3 Efficient training of the likelihood

Our goal is to directly optimize the likelihood in Eqn 3 which requires expensive linear algebra operations like inverting and calculating the log determinant of $A_\phi = RF_\theta R + \sigma_N^2 R$, for every $\phi = (\theta, \sigma^2)$ update. Since A_ϕ is symmetric, we could compute its log determinant and inverse using Cholesky. However, the computational cost of Cholesky is $\mathcal{O}(M^3)$, which given the size of M, would amount to a prohibitively expensive $\approx 10^{21}$ FLOPs per iteration!

Furthermore, in contrast to previous methods, we train neural network models for f_θ with millions of parameters, that is $\theta \in \mathbb{R}^D$ where $D \approx \mathcal{O}(10^6)$ and so computing $f_{\theta,m}$ for every variant m also becomes prohibitively expensive.

In order to circumvent the aforementioned costs of computing likelihood and to efficiently compute gradients for $\phi = (\theta, \sigma^2)$, we follow a two-pronged approach. First, we utilize a banded / sliding window approximation of R which allows us to amortize the training of θ across each slice and, second, we construct a specialized preconditioner which, in conjunction with fast iterative methods, allows us to efficiently optimize the likelihood.

Using submatrices for mini-batching Our first challenge is that calculating Eqn 3 requires us to compute $f_{\theta,m}$ for every m in the genome while performing expensive linear algebra operations on an enormous dense $M \times M$ matrix R.

First we make a standard approximation: we break the genome up into 2700 windows of size 1 million and assume the associations $\hat{\beta}$ in each window are generated independently. This can be justified by the fact that R is approximately block diagonal for instance (Berisa & Pickrell, 2016; Salehi Nowbandegani et al., 2023). Thus Eqn. 3 becomes

$$\sum_i \hat{\beta}_{(i)}^\intercal (A_\phi^{(i)})^{-1} \hat{\beta}_{(i)} + \log|A_\phi^{(i)}| \tag{5}$$

where $A_\phi^{(i)}$ is the submatrix of A_ϕ of variants in the i-th window and $\hat{\beta}_{(i)}$ is the subvector of $\hat{\beta}$ of variants in the i-th window.

Next note $A_\phi^{(i)} = R_{(i),:} F_\theta R_{:,(i)} + \sigma_N^2 R_{(i),(i)}$ where $R_{(i),:}$ represents the rectangular submatrix of R whose rows are variants in window i and $R_{(i),(i)}$ is similar. Calculating $A_\phi^{(i)}$ still requires

[1] LD score regression can also be derived in infinitesimal models more general than Bayesian linear models with a normal prior (Bulik-Sullivan et al., 2015).

calculating $f_{\theta,m}$ for every variant m. To avoid this calculation, we the use well established fact that variants that are distant in the genome should have little correlation, and so we can use a banded approximation of R (Bulik-Sullivan et al., 2015); in particular we assume that $R_{k,r} = 0$ when the positions of the k-th and r-th variants are more than 10^6 apart. Thus

$$A_\phi^{(i)} \approx R_{(i),(i)^+} F_\theta^{(i)} R_{(i)^+,(i)} + \sigma_N^2 R_{(i),(i)}$$

where $(i)^+$ is the set of all variants within 10^6 positions of a variant in window (i) and $F_\theta^{(i)}$ is the $(i)^+ \times (i)^+$ submatrix of F_θ. Now Eqn 5 then allows us to optimize ϕ, through stochastic gradient descent, by sampling windows (i) and only calculating $f_{\theta,m}$ for the roughly 10^4 variants in $(i)^+$. For simplicity, below we act as though $(i) = (i)^+$ and write $R^{(i)} = R_{(i),(i)}$.

Connection to LD score regression Due to the large size of our windows, we expect both approximations above to be accurate. In contrast, if we focus on the extreme case of a window size of 1 the objective Eqn. 5 becomes

$$\sum_i \frac{N\hat{\beta}_i^2}{N\sum_{m'} f_{m'} R_{mm'}^2 + \sigma^2} + \log(N \sum_{m'} f_{m'} R_{mm'}^2 + \sigma^2)$$

which tries to fit $N\sum_{m'} f_{m'} R_{mm'}^2 + \sigma^2$ to $N\hat{\beta}_i^2$. This is exactly the idea of LD score regression (Eqn. 4). Therefore LDSR can be thought of as our objective when assuming every $\hat{\beta}$ was generated independently.

Fast linear algebra with preconditioned iterative methods To train our models we not only have to invert and compute the log determinant of A_ϕ at every iteration but also backpropagate through these computations, increasing the complexity of the problem. Our approach is to use iterative methods like stochastic Lanczos quadrature (SLQ) (Golub & Loan, 2018; Saad, 2011) for $\log|A_\phi^{(i)}|$ and conjugate gradients (CG) (Nocedal & Wright, 2006; Golub & Loan, 2018; Saad, 2003) for solves $(A_\phi^{(i)})^{-1}$. Both methods perform multiplications against $A_\phi^{(i)}$ at each iteration, improving the quality of the approximation. Thus, the computational cost of both methods is a manageable $\mathcal{O}(JM_i^2)$ where J is the number of iterations. We review these methods in App. B.2.

The number of iterations required to converge below an error threshold of the iterative methods that we use is directly linked to the eigenspectrum of $A_\phi^{(i)}$ (Nocedal & Wright, 2006; Saad, 2011; Hogben, 2013). Therefore we can improve the convergence rate by finding a matrix P, known as a "preconditioner", such that $PA_\phi^{(i)} \approx I$ and replacing the liner algebra operations on $A_\phi^{(i)}$ with that of $PA_\phi^{(i)}$.

Before the construction of the WASP preconditioner we first have to pre-process the LD slices $R^{(i)}$. As opposed to $A_\phi^{(i)}$, which changes whenever we update ϕ, each $R^{(i)}$ is fixed throughout training. Therefore, before training, we compute the eigendecomposition of $R^{(i)} = V^{(i)}\Lambda^{(i)}(V^{(i)})^\intercal$ and zero out any negative eigenvalues in $\Lambda^{(i)}$, that is $\Lambda^{(i)} = \max(0, \Lambda^{(i)})$. As mentioned before, $R^{(i)}$ should in principle not have negative eigenvalues as it is psd. Yet, in practice, we most likely observe small numerical negative eigenvalues as a consequence of data inaccuracies.

Once we pre-processing step is done then we construct a preconditioner for $A_\phi^{(i)} = R^{(i)}F_\theta^{(i)}R^{(i)} + \sigma^2 R^{(i)}$ by approximating f_θ as a constant function: that is, $F_\theta^{(i)}$ is approximated by $\mu_\theta^{(i)} I$ where $\mu_\theta^{(i)} = \text{mean}(\text{diag}(F_\theta^{(i)}))$. We expect this approximation to be accurate especially when the first term in $A_\phi^{(i)}$ is small – $f_{\theta,m} << \sigma_N^2 = \Theta(1/N)$. Since $f_{\theta,m}$ is the expected effect from an individual variant, we expect it to usually be on the order $1/M << 1/N$. However, when our approximation is poor then the iterative algorithm will take longer to converge, but will still converge to the correct value. In practice, we must also regularize the matrix with ϵ; details are discussed in App. B.1.

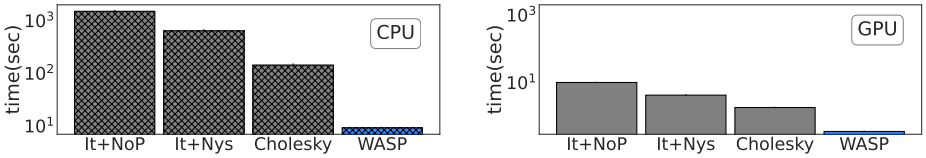

Figure 2: **WASP enables efficient loss and gradient computations.** We measure the time it takes to compute our loss which involves the computation of $(A_\phi^{(i)})^{-1}$ and $\log|A_\phi^{(i)}|$ as well as its gradients with respect to ϕ. We do this for 20 mini-batches of real UKBB data and display the mean runtime as barplots. `It` stands for the application of iterative algorithms such as SLQ and CG. We set a tolerance of 10^{-6} for CG and use 100 samples of SLQ. In terms of preconditioners, `NoP` implies that we did not use a preconditioner, `Nys` means that we used Nyström and `WASP` means that we applied our specialized structured preconditioner. For GPU we used an NVIDIA A100-SXM4-80GB and for CPU Intel(R) Xeon(R) Platinum 8268 CPU @ 2.90GHz.

Given the eigendecomposition of $R^{(i)}$ that we obtain through our pre-processing step, namely $(\Lambda^{(i)}, V^{(i)})$, we see that if we define the preconditioner P as

$$\begin{aligned} P^{-1} &= R^{(i)}(\mu_\theta I)R^{(i)} + \sigma_N^2 R^{(i)} + \epsilon I \\ &= V^{(i)}[\mu_\theta(\Lambda^{(i)})^2 + \sigma_N^2\Lambda^{(i)}](V^{(i)})^\intercal + \epsilon I \end{aligned}$$

then, using the Woodbury identity we get that $P = \frac{1}{\epsilon}I - \frac{1}{\epsilon}V^{(i)}\Gamma_\phi^{(i)}(V^{(i)})^\intercal$ where $\Gamma_\phi^{(i)}$ is a diagonal matrix such that $\Gamma_{k,k}^{(i)} = \frac{1}{\mu_\theta(\lambda_k^{(i)})^2 + \sigma_N^2\lambda_k^{(i)}} + \epsilon$. Note that the construction of P comes at almost no cost, as we only have to compute, at each iteration, μ_θ and $\Gamma_\phi^{(i)}$ to update P.

We demonstrate the efficacy of our method in 2 where we see how WASP significantly reduces the runtime when compared to other methods like Cholesky and other preconditioning strategies commonly used for other large scale Bayesian models as in Gardner et al. (2018) or Frangella et al. (2021).

4 Predicting variant effects from functional annotations

Now we have a method for accurately and efficiently training a model f_θ. Here we specify how we build f_θ that include many more functional and comparative genomics features than previous works.

Features Previous methods have built f_θ using functional genomics features such as DNA accessibility, proximity to functional elements, and presence of a coding region and comparative genomics features such as conservation scores (Finucane et al., 2015; Li et al., 2024). Many of these features are defined as annotations at each position in the genome; to get a single value, annotations were averaged in a window before being passed to f_θ.

We expand this set in two ways. First we consider a significantly expanded set of functional genomics annotations – binding and accessibility annotations from ENCODE (ENCODE Project Consortium, 2012), enhancer annotations from FANTOM (Lizio et al., 2015) – and comparative genomics annotations – conservation scores such as PhyloP (Pollard et al., 2010), and predictions of effects of mutations in coding regions such as those from ESM2 (Liu et al., 2020). Details of these data are in App. C.2 and C.3. Second, instead of considering an average of the values of annotations in a window around the variant, we pass the model the exact values of the annotations at all positions in the window.

Gazal et al. (2017) used LDSR demonstrated that the recent history of a variant in humans can also be predictive of its effect size. To account for this, we also included the frequency of each variant m, freq_m; its "minor allele" frequency, $\min\{\text{freq}_m, 1 - \text{freq}_m\}$; and its LD score $\sum_{m'} R_{mm'}^2$ as features.

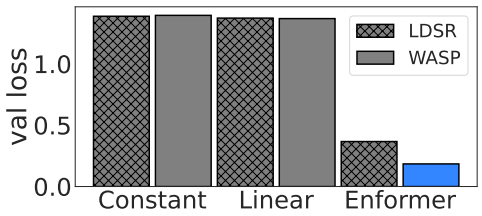

Figure 3: **WASP using an enformer model best recovers f_θ.** The bars represent the RMSE difference between the learnt $\hat{f}_\theta$ and the ground truth f_θ in the log space evaluated over a set of validation tracks (lower is better).

Architecture For all genomics annotations but coding mutation effect predictions, we consider a window around each variant of size w. We pass these tracks $C_{\text{track},m} \in \mathbb{R}^{d_{\text{track}} \times w}$ along with predictions of the effects of mutations if the mutation is in a coding region and genomic architecture information, $C_{\text{pred},m} \in \mathbb{R}^{d_{\text{pred}}}$, to a neural network $f_\theta(C_{\text{track},m}, C_{\text{pred},m})$. In our case, $d_{\text{track}} = 165$, and $d_{\text{pred}} = 9$; we also choose a window size of $w = 256$. The architecture we use is adapted from a network used to predict tracks from sequence, Enformer (Avsec et al., 2021); this architecture uses a mix of convolutional and attention layers.

Speed et al. (2017) suggested that setting $f = \text{constant}$ in our model makes the implicit assumption that rare variants have larger effects; They generalized our model to remove this assumption with a more general model $f_m = (\text{freq}_m(1 - \text{freq}_m))^\alpha$ where α is a fit parameter. In our case, we consider

$$f_{\theta,m} = (\text{freq}_m(1 - \text{freq}_m))^\alpha \text{NN}_\theta(C_{\text{track},m}, C_{\text{pred},m}) \tag{6}$$

where NN_θ is a neural network or any other model. In our experiments below, α typically converges to a value between 0.6 and 0.7 regardless of its initialization.

5 Empirical Results

We now apply WASP to train flexible neural network models in order to better explain which variants are associated with phenotypic traits. See B for experimental details.

In all of our experiments we use public data from the UK Biobank (UKBB) of over 300,000 European individuals. We download LD matrices R calculated from (Weissbrod et al., 2020). We use the $\hat{\beta}$s from 3 traits calculated in Loh et al. (2015) – body mass index, height, and asthma. See App. C.1 for details.

Semi-synthetic simulation Semi-synthetic simulations are a staple of statistical genetics literature (ex. Candès et al. (2016) or O'Connor et al. (2019)) and are used to validate methods when true effects β are unavailable. We use them to demonstrate that our loss function and our numerical techniques allow us to recover different f_θ functions while using real LD patterns. That is, we use the real R provided by UKBB but generate $\hat{\beta}$ as follows:

$$\beta_m \sim \mathcal{N}\left(0, f_m\left(C_{\text{track},m}, C_{\text{pred},m}\right)\right); \hat{\beta} \sim \mathcal{N}\left(R\beta, \sigma_N R\right)$$

where f represents that function that we are trying to learn. We consider f_θ as a randomly initialized Enformer neural network model (16 million parameters). See App. B.7 for details.

We tried fitting this data with models based on Eqn. 6. We used simple models – $\text{NN}_\theta = \text{constant}$ and $\text{NN}_\theta =$ generalized linear model (see App. B.4) – and a more flexible NN_θ with the enformer architecture with LD score regression (LDSR) and WASP. In Figure 3 we show WASP with a large model can closely recover the true variant effect distribution f_θ – it achieves a low error in predictions f_θ. Furthermore, this model better predictions than restricted constant and linear f_θ. We also see that our method makes more accurate predictions than models trained with LD score regression.

Method	Model	BMI	Height	Asthma
LDSR	Constant	524	1619	27.2
LDSR	Linear	568	1880	50.0
LDSR	Enformer	539	1692	8.9
WASP	Constant	520	1560	30.4
WASP	Linear	584	1916	58.1
WASP	Enformer	**635**	**2070**	**66.4**

Table 1: **More flexible models trained with WASP better explain genetic associations.** We report the increase in likelihood on the test chromosomes over a null model ($f = 0$).

Model (params)	BMI	Height	Asthma
Reduced features	608	2032	64.5
Reduced model size	629	2015	61.8
Full model	**635**	**2070**	**66.4**

Table 2: **Larger models with more features better explain genetic associations.** We ablate the feature set and model size of our enformer model. We report the increase in likelihood on the test chromosomes over a null model ($f = 0$).

Fitting association data on UKBB Now we use WASP to explain the associations of variants to real phenotypes in UKBB. We quantify how well each model explains associations with a trait using the difference in likelihood Eqn. 3 using our learned f_θ versus a null model with no heritable effect $f_\theta = 0$. We train on associations from variants in chromosomes 1-20 and evaluate our model on holdout variants in chromosomes 21 and 22.

We first evaluated the likelihoods of various architectures trained with LDSR and WASP on held-out chromosomes against a null model. LDSR and WASP used similar computational resources in training. Table 1 shows that a model with constant f better explains the data than a null model, $f = 0$, and a generalized linear model in turn better explains the data than a constant model. Training these small models with LDSR or WASP resulted in similar or slightly improved quality models.

In contrast, when we train Enformer with LDSR, it surprisingly performs worse than the linear model. The more flexible architecture potentially over-fits the data due to the loss of statistical efficiency when performing LDSR. When accounting for correlations in the $\hat{\beta}$ with WASP however, the enformer substantially outperform all other models.

We were next interested to determine how important the feature set and model size are for the performance of our models. We trained models with a reduced feature set – we looked at a window of $w = 128$ rather than $w = 256$ around each variant, and removed the 111 features from ENCODE – and with a reduced size – we reduce the number of parameters from 16 million to 4.4 million. Table 2 shows that ablating the model size or feature set often harms model performance. This degradation strongly suggests that our model benefits from its flexibility and features to better predict the effects of variants.

6 Conclusion

By efficiently inverting LD matrices, WASP allows us to train large models that better predict the effects of variants on phenotype and to learn their functional causes. Our results demonstrate that larger models make better predictions than the simple models used in practice, and that increasing the model size and and using more features improves predictive power.

Meaningfulness Statement WASP learns representations of functional genomics data that are useful for predicting disease. This is distinct from the representations learned in other predictive models.

References

Ziga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *bioRxiv*, pp. 2021.04.07.438649, 2021.

Nick H Barton, Alison M Etheridge, and Amandine Véber. The infinitesimal model. *bioRxiv*, February 2016.

Tomaz Berisa and Joseph K Pickrell. Approximately independent linkage disequilibrium blocks in human populations. *Bioinformatics*, 32(2):283–285, January 2016.

Evan A Boyle, Yang I Li, and Jonathan K Pritchard. An expanded view of complex traits: From polygenic to omnigenic. *Cell*, 169(7):1177–1186, June 2017.

Brendan K Bulik-Sullivan, Po-Ru Loh, Hilary K Finucane, Stephan Ripke, Jian Yang, Schizophrenia Working Group of the Psychiatric Genomics Consortium, Nick Patterson, Mark J Daly, Alkes L Price, and Benjamin M Neale. LD score regression distinguishes confounding from polygenicity in genome-wide association studies. *Nat. Genet.*, 47(3): 291–295, March 2015.

E Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80, October 2016.

Ashley Mae Conard, Alan DenAdel, and Lorin Crawford. A spectrum of explainable and interpretable machine learning approaches for genomic studies. *Wiley Interdiscip. Rev. Comput. Stat.*, May 2023.

Gregory M Cooper, Eric A Stone, George Asimenos, NISC Comparative Sequencing Program, Eric D Green, Serafim Batzoglou, and Arend Sidow. Distribution and intensity of constraint in mammalian genomic sequence. *Genome Res.*, 15(7):901–913, July 2005.

ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, September 2012.

Tabassum Fabiha, Ivy Evergreen, Soumya Kundu, Anusri Pampari, Sergey Abramov, Alexandr Boytsov, Kari Strouse, Katherine Dura, Weixiang Fang, Gaspard Kerner, John Butts, Thahmina Ali, Andreas Gschwind, Kristy S Mualim, Jill E Moore, Zhiping Weng, Jacob Ulirsch, Hongkai E Ji, Jeff Vierstra, Timothy E Reddy, Stephen B Montgomery, Jesse Engreitz, Anshul Kundaje, Ryan Tewhey, Alkes Price, and Kushal Dey. A consensus variant-to-function score to functionally prioritize variants for disease. *bioRxiv*, November 2024.

Hilary K Finucane, Brendan Bulik-Sullivan, Alexander Gusev, Gosia Trynka, Yakir Reshef, Po-Ru Loh, Verneri Anttila, Han Xu, Chongzhi Zang, Kyle Farh, Stephan Ripke, Felix R Day, ReproGen Consortium, Schizophrenia Working Group of the Psychiatric Genomics Consortium, RACI Consortium, Shaun Purcell, Eli Stahl, Sara Lindstrom, John R B Perry, Yukinori Okada, Soumya Raychaudhuri, Mark J Daly, Nick Patterson, Benjamin M Neale, and Alkes L Price. Partitioning heritability by functional annotation using genome-wide association summary statistics. *Nat. Genet.*, 47(11):1228–1235, November 2015.

Zachary Frangella, Joel A. Tropp, and Madeleine Udell. Randomized Nyström Preconditioning. *arXiv 2110.02820v2*, 2021.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. GPyTorch: blackbox matrix-matrix gaussian process inference with GPU acceleration. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7587–7597, Red Hook, NY, USA, December 2018. Curran Associates Inc.

Steven Gazal, Hilary K Finucane, Nicholas A Furlotte, Po-Ru Loh, Pier Francesco Palamara, Xuanyao Liu, Armin Schoech, Brendan Bulik-Sullivan, Benjamin M Neale, Alexander Gusev, and Alkes L Price. Linkage disequilibrium-dependent architecture of human complex traits shows action of negative selection. *Nat. Genet.*, 49(10):1421–1427, October 2017.

Gene H Golub and Charles F Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2018. Fourth Edition.

Alexander Gusev, S Hong Lee, Gosia Trynka, Hilary Finucane, Bjarni J Vilhjálmsson, Han Xu, Chongzhi Zang, Stephan Ripke, Brendan Bulik-Sullivan, Eli Stahl, Schizophrenia Working Group of the Psychiatric Genomics Consortium, SWE-SCZ Consortium, Anna K Kähler, Christina M Hultman, Shaun M Purcell, Steven A McCarroll, Mark Daly, Bogdan Pasaniuc, Patrick F Sullivan, Benjamin M Neale, Naomi R Wray, Soumya Raychaudhuri, Alkes L Price, Schizophrenia Working Group of the Psychiatric Genomics Consortium, and SWE-SCZ Consortium. Partitioning heritability of regulatory and cell-type-specific variants across 11 common diseases. *Am. J. Hum. Genet.*, 95(5):535–552, November 2014.

Bjarni V Halldorsson, Hannes P Eggertsson, Kristjan H S Moore, Hannes Hauswedell, Ogmundur Eiriksson, Magnus O Ulfarsson, Gunnar Palsson, Marteinn T Hardarson, Asmundur Oddsson, Brynjar O Jensson, Snaedis Kristmundsdottir, Brynja D Sigurpalsdottir, Olafur A Stefansson, Doruk Beyter, Guillaume Holley, Vinicius Tragante, Arnaldur Gylfason, Pall I Olason, Florian Zink, Margret Asgeirsdottir, Sverrir T Sverrisson, Brynjar Sigurdsson, Sigurjon A Gudjonsson, Gunnar T Sigurdsson, Gisli H Halldorsson, Gardar Sveinbjornsson, Kristjan Norland, Unnur Styrkarsdottir, Droplaug N Magnusdottir, Steinunn Snorradottir, Kari Kristinsson, Emilia Sobech, Gudmar Thorleifsson, Frosti Jonsson, Pall Melsted, Ingileif Jonsdottir, Thorunn Rafnar, Hilma Holm, Hreinn Stefansson, Jona Saemundsdottir, Daniel F Gudbjartsson, Olafur T Magnusson, Gisli Masson, Unnur Thorsteinsdottir, Agnar Helgason, Hakon Jonsson, Patrick Sulem, and Kari Stefansson. The sequences of 150,119 genomes in the UK biobank. *bioRxiv*, pp. 2021.11.16.468246, November 2021.

Leslie Hogben. Handbook of Linear Algebra. *Chapman and Hall/CRC*, 2013.

Farhad Hormozdiari, Emrah Kostem, Eun Yong Kang, Bogdan Pasaniuc, and Eleazar Eskin. Identifying causal variants at loci with multiple signals of association. *Genetics*, 198(2): 497–508, October 2014.

Kexin Huang, Tony Zeng, Soner Koc, Alexandra Pettet, Jingtian Zhou, Mika Jain, Dongbo Sun, Camilo Ruiz, Hongyu Ren, Laurence Howe, Tom G Richardson, Adrian Cortes, Katie Aiello, Kim Branson, Andreas Pfenning, Jesse M Engreitz, Martin Jinye Zhang, and Jure Leskovec. Small-cohort GWAS discovery with AI over massive functional genomics knowledge graph. *Genetic and Genomic Medicine*, (medrxiv;2024.12.03.24318375v1), December 2024.

Melissa J Hubisz, Katherine S Pollard, and Adam Siepel. PHAST and RPHAST: phylogenetic analysis with space/time models. *Brief. Bioinform.*, 12(1):41–51, January 2011.

Gleb Kichaev, Gaurav Bhatia, Po-Ru Loh, Steven Gazal, Kathryn Burch, Malika K Freund, Armin Schoech, Bogdan Pasaniuc, and Alkes L Price. Leveraging polygenic functional enrichment to improve GWAS power. *Am. J. Hum. Genet.*, 104(1):65–75, January 2019.

Hui Li, Tushar Kamath, Rahul Mazumder, Xihong Lin, and Luke O'Connor. Improved heritability partitioning and enrichment analyses using summary statistics with graphREML. *Genetic and Genomic Medicine*, (medrxiv;2024.11.04.24316716v1), November 2024.

Xihao Li, Zilin Li, Hufeng Zhou, Sheila M Gaynor, Yaowu Liu, Han Chen, Ryan Sun, Rounak Dey, Donna K Arnett, Stella Aslibekyan, Christie M Ballantyne, Lawrence F Bielak, John Blangero, Eric Boerwinkle, Donald W Bowden, Jai G Broome, Matthew P Conomos, Adolfo Correa, L Adrienne Cupples, Joanne E Curran, Barry I Freedman, Xiuqing Guo, George Hindy, Marguerite R Irvin, Sharon L R Kardia, Sekar Kathiresan, Alyna T Khan, Charles L Kooperberg, Cathy C Laurie, X Shirley Liu, Michael C Mahaney,

Ani W Manichaikul, Lisa W Martin, Rasika A Mathias, Stephen T McGarvey, Braxton D Mitchell, May E Montasser, Jill E Moore, Alanna C Morrison, Jeffrey R O'Connell, Nicholette D Palmer, Akhil Pampana, Juan M Peralta, Patricia A Peyser, Bruce M Psaty, Susan Redline, Kenneth M Rice, Stephen S Rich, Jennifer A Smith, Hemant K Tiwari, Michael Y Tsai, Ramachandran S Vasan, Fei Fei Wang, Daniel E Weeks, Zhiping Weng, James G Wilson, Lisa R Yanek, NHLBI Trans-Omics for Precision Medicine (TOPMed) Consortium, TOPMed Lipids Working Group, Benjamin M Neale, Shamil R Sunyaev, Gonçalo R Abecasis, Jerome I Rotter, Cristen J Willer, Gina M Peloso, Pradeep Natarajan, and Xihong Lin. Dynamic incorporation of multiple in silico functional annotations empowers rare variant association analysis of large whole-genome sequencing studies at scale. *Nat. Genet.*, 52(9):969–983, September 2020.

Xiaoming Liu, Chang Li, Chengcheng Mou, Yibo Dong, and Yicheng Tu. dbNSFP v4: a comprehensive database of transcript-specific functional predictions and annotations for human nonsynonymous and splice-site SNVs. *Genome Med.*, 12(1):103, December 2020.

Marina Lizio, Jayson Harshbarger, Hisashi Shimoji, Jessica Severin, Takeya Kasukawa, Serkan Sahin, Imad Abugessaisa, Shiro Fukuda, Fumi Hori, Sachi Ishikawa-Kato, Christopher J Mungall, Erik Arner, J Kenneth Baillie, Nicolas Bertin, Hidemasa Bono, Michiel de Hoon, Alexander D Diehl, Emmanuel Dimont, Tom C Freeman, Kaori Fujieda, Winston Hide, Rajaram Kaliyaperumal, Toshiaki Katayama, Timo Lassmann, Terrence F Meehan, Koro Nishikata, Hiromasa Ono, Michael Rehli, Albin Sandelin, Erik A Schultes, Peter A C 't Hoen, Zuotian Tatum, Mark Thompson, Tetsuro Toyoda, Derek W Wright, Carsten O Daub, Masayoshi Itoh, Piero Carninci, Yoshihide Hayashizaki, Alistair R R Forrest, Hideya Kawaji, and FANTOM consortium. Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biol.*, 16(1):22, January 2015.

Po-Ru Loh, George Tucker, Brendan K Bulik-Sullivan, Bjarni J Vilhjálmsson, Hilary K Finucane, Rany M Salem, Daniel I Chasman, Paul M Ridker, Benjamin M Neale, Bonnie Berger, Nick Patterson, and Alkes L Price. Efficient bayesian mixed-model analysis increases association power in large cohorts. *Nat. Genet.*, 47(3):284–290, March 2015.

Po-Ru Loh, Gleb Kichaev, Steven Gazal, Armin P Schoech, and Alkes L Price. Mixed-model association for biobank-scale datasets. *Nat. Genet.*, 50(7):906–908, July 2018.

Qiongshi Lu, Xinwei Yao, Yiming Hu, and Hongyu Zhao. GenoWAP: GWAS signal prioritization through integrated analysis of genomic functional annotation. *Bioinformatics*, 32 (4):542–548, February 2016.

Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, Judy H Cho, Alan E Guttmacher, Augustine Kong, Leonid Kruglyak, Elaine Mardis, Charles N Rotimi, Montgomery Slatkin, David Valle, Alice S Whittemore, Michael Boehnke, Andrew G Clark, Evan E Eichler, Greg Gibson, Jonathan L Haines, Trudy F C Mackay, Steven A McCarroll, and Peter M Visscher. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, October 2009.

Guiyan Ni, Gerhard Moser, Schizophrenia Working Group of the Psychiatric Genomics Consortium, Naomi R Wray, and S Hong Lee. Estimation of genetic correlation via linkage disequilibrium score regression and genomic restricted maximum likelihood. *Am. J. Hum. Genet.*, 102(6):1185–1194, June 2018.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006. Second Edition.

Luke J O'Connor, Armin P Schoech, Farhad Hormozdiari, Steven Gazal, Nick Patterson, and Alkes L Price. Extreme polygenicity of complex traits is explained by negative selection. *Am. J. Hum. Genet.*, 105(3):456–476, September 2019.

Katherine S Pollard, Melissa J Hubisz, Kate R Rosenbloom, and Adam Siepel. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.*, 20(1):110–121, January 2010.

Andres Potapczynski, Marc Finzi, Geoff Pleiss, and Andrew Gordon Wilson. CoLA: Exploiting Compositional Structure for Automatic and Efficient Numerical Linear Algebra. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yousef Saad. Iterative Methods for Sparse Linear Systems. *SIAM*, 2003.

Yousef Saad. Numerical methods for large eigenvalue problems. *SIAM*, 2011.

Pouria Salehi Nowbandegani, Anthony Wilder Wohns, Jenna L Ballard, Eric S Lander, Alex Bloemendal, Benjamin M Neale, and Luke J O'Connor. Extremely sparse models of linkage disequilibrium in ancestrally diverse association studies. *Nat. Genet.*, 55(9): 1494–1502, September 2023.

Huwenbo Shi, Gleb Kichaev, and Bogdan Pasaniuc. Contrasting the genetic architecture of 30 complex traits from summary association data. *Am. J. Hum. Genet.*, 99(1):139–153, July 2016.

Doug Speed, Na Cai, UCLEB Consortium, Michael R Johnson, Sergey Nejentsev, and David J Balding. Reevaluation of SNP heritability in complex human traits. *Nat. Genet.*, 49(7): 986–992, July 2017.

Jeffrey P Spence, Nasa Sinnott-Armstrong, Themistocles L Assimes, and Jonathan K Pritchard. A flexible modeling and inference framework for estimating variant effect sizes from GWAS summary statistics. *bioRxiv*, April 2022.

Brian Trippe, Hilary Finucane, and Tamara Broderick. For high-dimensional hierarchical models, consider exchangeability of effects across covariates instead of across datasets. In M Ranzato, A Beygelzimer, Y Dauphin, P S Liang, and J Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13471–13484. Curran Associates, Inc., 2021.

Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 10 years of GWAS discovery: Biology, function, and translation. *Am. J. Hum. Genet.*, 101(1):5–22, July 2017.

Omer Weissbrod, Farhad Hormozdiari, Christian Benner, Ran Cui, Jacob Ulirsch, Steven Gazal, Armin P Schoech, Bryce van de Geijn, Yakir Reshef, Carla Márquez-Luna, Luke O'Connor, Matti Pirinen, Hilary K Finucane, and Alkes L Price. Functionally informed fine-mapping and polygenic localization of complex trait heritability. *Nat. Genet.*, 52(12): 1355–1363, December 2020.

Jian Yang, Beben Benyamin, Brian P McEvoy, Scott Gordon, Anjali K Henders, Dale R Nyholt, Pamela A Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, Michael E Goddard, and Peter M Visscher. Common SNPs explain a large proportion of the heritability for human height. *Nat. Genet.*, 42(7):565–569, July 2010.

Qianqian Zhang, Florian Privé, Bjarni Vilhjálmsson, and Doug Speed. Improved genetic prediction of complex traits from individual-level data or summary statistics. *Nat. Commun.*, 12(1):4192, July 2021.

Zhili Zheng, Shouye Liu, Julia Sidorenko, Ying Wang, Tian Lin, Loic Yengo, Patrick Turley, Alireza Ani, Rujia Wang, Ilja M Nolte, Harold Snieder, LifeLines Cohort Study, Jian Yang, Naomi R Wray, Michael E Goddard, Peter M Visscher, and Jian Zeng. Leveraging functional genomic annotations and genome coverage to improve polygenic prediction of complex traits within and between ancestries. *Nat. Genet.*, 56(5):767–777, May 2024.

A Detailed previous work

Training functionally informed priors Training a functionally informed prior by directly optimizing the likelihood of the data has, up until now, been computationally prohibitive due to the cost of linear algebra operations on the LD matrix. Previous methods have used

a number of strategies to restrict the flexibility of their prior or looked at other approximate or derived objectives in order to do inference. First, most GWAS methods pick their prior with only a handful of parameters (usually 1 or 2) and fit it by grid search or other bespoke methods that struggle to scale Yang et al. (2010); Loh et al. (2015); Speed et al. (2017); Spence et al. (2022). Second, Finucane et al. (2015) fit a linear prior by performing LD score regression in Eqn. 4. Third, Lu et al. (2016) and Fabiha et al. (2024) fit a small model by teaching it to classify the small number (≈ 2000) of available high confidence positive and negative causal variants. Fourth, Li et al. (2024) considered fitting a simple generalized linear model f_θ by approximating Eqn. 3 using an approximation of R^{-1}. All of these methods run on CPU and use parallelism to compute the gradient of the likelihood across the entire genome for each update.

Unfortunately these methods are unsuitable for training a large flexible prior as they 1) lose statistical power by approximating the likelihood and 2) they require prior values for all variants in the genome for a single gradient update. In contrast, our method WASP 1) directly optimizes the likelihood of data from millions of variants, 2) updates the model using its predictions in minibatches, and accelerates linear algebra operations in each minibatch with GPUs.

In related work Huang et al. (2024) fit a graph neural network of variants to predict $\hat{\beta}$ directly; they use their model to increase power to find more associated variants. However such a model does not distinguish between variants with large effects β and variants they are associated with.

Downstream uses of functionally informed priors A number of works have built methods to use functionally informed priors to increase the power of downstream analyses. Huang et al. (2024) and Kichaev et al. (2019) demonstrated that models that can predict the effect of variants can improve the power of GWAS. Weissbrod et al. (2020) demonstrated such models can also identify causal variants and Li et al. (2020) used such variants to identify causal genes. The WASP prior can in principle fit into these same pipelines.

Flexible models of heritability In addition to more flexible models predicting variant effects from functional annotations, we can improve fits to association data with models that are more flexible than mixed linear models. Zhang et al. (2021) consider different, non-normal, priors, and Loh et al. (2015) consider mixture of normal priors on the effect sizes. There have also been a number of nonlinear models for predicting y from X (Conard et al., 2023). For simplicity, WASP considers the popular normal prior with a linear model and leaves more flexible models to future work.

Fast linear algebra with large genotype matrices A number of other works have looked at approximately inverting matrices of genetic variants to accelerate variant effect prediction. Loh et al. (2015) used a conjugate gradient algorithm to invert the matrix of correlations of variants between study individuals – the empirical kinship matrix $XX^\intercal$; Loh et al. (2018) noted that their algorithm converges much faster after removing the top eigenvalues of the kinship matrix , improving it condition number. Berisa & Pickrell (2016) approximated R with a block diagonal matrix, Shi et al. (2016) approximated R with a low rank matrix, and Salehi Nowbandegani et al. (2023) approximated the inverse of the R with an extremely sparse matrix; these works use these approximations in place of the true R. WASP uses an iterative algorithm to perform linear algebra operations on the exact R; we also build an approximation of the matrix we wish to invert but use this approximation to speed up linear algebra operations by using it as a preconditioner.

Fast linear algebra for fitting large Bayesian models Fitting Gaussian processes similarly involves inverting a large matrix known as the Gram matrix. While one can avoid inverting the matrix with variational inference, state of the art methods invert the Gram matrix with an iterative algorithm with a Nyström preconditioner (Gardner et al., 2018). We build a bespoke preconditioner leveraging the structure of LD matrices to quickly invert LD matrices with iterative algorithms; in our setting, our preconditioner performs much better than a general purpose Nyström preconditioner (Frangella et al., 2021).

B Experimental details

B.1 Regularizing the submatrices

The main properties that characterize $R^{(i)}$ is that it is positive semi-definite (psd) and that it is singular since several traits are highly correlated with each other. By construction, these two properties are also inherited by $A_\phi^{(i)}$ and so the main numerical challenge when optimizing Eqn 5 is that we need to deal with the fact that $R^{(i)}$ or $A_\phi^{(i)}$ are singular.

B.2 Iterative algorithms

In terms of implementation, we use `CoLA` (Potapczynski et al., 2023), a numerical linear algebra library that is compatible with diverse deep learning frameworks and that provides backpropagation capabilities for SLQ and CG. `CoLA` computes the gradients of A_ϕ^{-1} and $\log |A_\phi^{-1}|$ by using the following identities

$$\nabla_\phi A_\phi^{-1} = -A_\phi^{-1} \nabla_\phi A_\phi A_\phi^{-1}$$
$$\nabla_\phi \log \left| A_\phi^{-1} \right| = \text{trace}(A_\phi^{-1} \nabla_\phi A_\phi) = \mathbb{E}_{u \sim \mathcal{N}(0,I)} (A_\phi^{-1} u)^\intercal \nabla_\phi A_\phi u$$

where both quantities require backpropagating through A_ϕ only and where we use the Hutchinson trace estimator. Additionally, `CoLA` allows us to leverage GPU acceleration for our numerical techniques which significantly reduces the runtime.

Previous works like Salehi Nowbandegani et al. (2023) or Hormozdiari et al. (2014), deal with the singularity issues by adding regularization to $R^{(i)}$ as $R^{(i)} + \epsilon I$ for some small ϵ. The problem with this approach is that, in our case, the regularization ϵ gets affected by the scale of σ_N^2 and influenced by θ since $(R^{(i)} + \epsilon I) F_\theta^{(i)} (R^{(i)} + \epsilon I) + \sigma_N^2 (R^{(i)} + \epsilon I) = A_\phi^{(i)} + \epsilon R F_\theta^{(i)} + \epsilon F_\theta^{(i)} R + \epsilon^2 F_\theta^{(i)} + \epsilon \sigma_N^2 I$. It thus becomes unclear how close we are to the original problem if the regularization keeps changing at each iteration.

In contrast, we choose to add the regularization directly to $A_\phi^{(i)}$ as, $A_\phi^{(i)} + \epsilon I$ and leave $R^{(i)}$ untouched. In our experiments we set $\epsilon \approx 10^{-4}$.

B.3 Models

We obtained code for enformer from `https://github.com/lucidrains/enformer-pytorch` under the MIT license. We reduce the internal dimension to 768 and the number of transformer layers to 2. Our "smaller model" further reduced the internal dimension to 384.

We normalize features to have mean 0 and variance 1 across the genome before passing them to any model.

B.4 Generalized linear model

As a baseline we consider a generalized linear model as suggested in Li et al. (2024) using averages of each track in the window as in Finucane et al. (2015):

$$f_{\theta,m} = (\text{freq}_m(1 - \text{freq}_m))^\alpha \left(\sum_d w_d \sum_w C_{\text{track},m,d,w} + \sum_{d'} w'_{d'} C_{\text{pred},m,d'} \right) + c$$

where $(w_d)_d$, $(w'_{d'})_{d'}$, and c are learnable parameters.

B.5 Training

We trained our models with an AdamW optimizer with default hyperparameters, 100 warmup steps with a linear schedule. For σ and α we used a learning rate of 0.0002; for

θ we use a learning rate of 0.0001 for enformer models and 0.002 for linear and constant models. We train enformer models for up to 10 epochs; we trained smaller models for 2 epochs. We train models on single A100 or H100 GPUS on an academic cluster; enformer models were trained for 10 to 12 hours.

B.6 LD score regression (LDSR)

Finucane et al. (2015) suggested performing the linear LD score regression with a square loss 1) dividing by the (rough) standard deviation of a chi-squared variable and 2) down-weighting variants in LD with many other variants: calling $l = \mathbb{1}^T R^{\circ 2}$ and $h_g^2 = E_i f_{\theta,i}$ (ballpark estimate made before training), we minimize (we also multiply numerator and denominator by N)

$$\sum \frac{1}{l_i} \frac{1}{(N h_g^2 l_i / M + 1)^2} \left(\frac{N}{M} R_i^{\circ 2} f_\theta + \sigma^2 - N \hat{\beta}_i^2 \right)^2 .$$

B.7 Simulation

Here we describe how we chose a realistic f for semi-synthetic simulation. Recall,

$$y \sim N(0, \frac{1}{N} X^T F X + \sigma^2 I).$$

Therefore

$$1 = \mathrm{Var}(y_{i,i}) = \sigma^2 + \frac{1}{N} \sum_m X_{m,i}^2 F_m.$$

Assuming presence of a variant $X_{m,i}$ is independent of F_m, we have

$$1 = \mathrm{Var}(y_{i,i}) \approx \sigma^2 + \frac{M}{N} E_m[X_{m,i}^2] E_m[F_m] = \sigma^2 + \frac{M}{N} E_m[F_m].$$

Thus, in our simulated data, ideally we would ensure that

$$E_m F_m = \frac{N}{M}(1 - \sigma^2).$$

In our case, we choose a highly heritable disease with $\sigma^2 = 1/2$ so half of the variance of y is from the noise ϵ and the other half is genetic. Using real values $N = 407527$ and $M = 11904924$ for our data, we set $E_m f_m = \frac{N}{2M}$ by initializing a $\tilde{f}$, calculating $E_m \tilde{f}_m$, and defining $f_m = \frac{N}{2M E_m \tilde{f}_m} \tilde{f}_m$.

We defined $\tilde{f}_m = \exp(10 \times \mathrm{NN}_\theta(C_{\mathrm{track},m}, C_{\mathrm{pred},m}))$ where NN_θ is a randomly initialized Enformer model.

C Data Collection

C.1 UKBB summary statistics

We downloaded UK biobank LD matrices computed in Weissbrod et al. (2020) from the Amazon web services S3 container `s3://broad-alkesgroup-ukbb-ld/UKBB_LD/`. These matrices can have small negative eigenvalues, which we removed prior to training.

We downloaded UK biobank association statistics computed using BOLT-LMM (Loh et al., 2015) from the UKBB_409K folder in `https://console.cloud.google.com/storage/browser/broad-alkesgroup-public-requester-pays`. These association statistics also contained frequencies of each variant. Any variants that have LD information but that are missing associations are discarded; all variants with association information also had LD information.

UKBB coordinates are in GrCh37 but many of our features below are in the GrCh38 build. We used rsid's and pyliftOver (`https://github.com/konstantint/pyliftover`) to map to GrCh38. For the handful of variants we could not map we gave them the location of a nearby variant.

C.2 Coding variant annotations

We downloaded the predictions of the effects of variants in coding regions from various models from `https://www.dbnsfp.org/`. We used six predictions labeled `ESM1b_score`, `GERP++_RS`, `SIFT_score`, `PROVEAN_score`, `FATHMM_score`, `EVE_score`. For non-coding variants or variants missing a prediction, we set $C = 0$.

C.3 Functional and conservation track data

Conservation We downloaded bigWig files of our phylogenetic correlation tracks from `http://hgdownload.soe.ucsc.edu/goldenPath/hg38/` (Pollard et al., 2010; Hubisz et al., 2011). We used 15 PhyloP and phastCons scores made from various alignments: `phyloP470way`, `phyloP447way`, `phyloP100way`, `phyloP30way`, `phyloP20way`, `phyloP17way`, `phyloP7way`, `phyloP4way`, `phastCons470way`, `phastCons100way`, `phastCons30way`, `phastCons20way`, `phastCons17way`, `phastCons7way`, `phastCons4way`.

FANTOM We downloaded hCAGE FANTOM tracks of human tissues from `https://fantom.gsc.riken.jp/5/datahub/hg38/tpm/human.tissue.hCAGE/` (Lizio et al., 2015). This gave us roughly 400 tracks; we picked a random 20 tissues from this set and collected forward and backward CAGE tracks for each tissue, giving us a total of 40 features. The tissues were `lymph node, adult, donor1; heart, adult, diseased post-infarction, donor1; skeletal muscle, adult, pool1; occipital lobe, adult, donor1; parietal cortex, adult, donor10258; thymus, adult, pool1; thyroid, adult, pool1; pons, adult, pool1; parotid gland, adult; Fingernail (including nail plate, eponychium and hyponychium), donor2; thalamus, adult, donor10258; caudate nucleus, adult, donor10252; parietal lobe, adult, donor10252; cerebrospinal fluid, donor2; kidney, fetal, pool1; eye - muscle inferior rectus, donor1; nucleus accumbens, adult, pool1; parietal lobe - adult, donor10196; cerebral meninges, adult; throat, adult.`

ENCODE We downloaded bigWig files of functional genomics tracks from `https://www.encodeproject.org/search/` (ENCODE Project Consortium, 2012). We did not use tracks with warnings, errors, or that were non-compliant. We used assays with titles `TF ChIP-seq`, `Histone ChIP-seq`, `eCLIP`, `total RNA-seq`, `polyA plus RNA-seq`, `polyA minus RNA-seq`, `small RNA-seq`, `microRNA-seq`, `ChIA-PET`, `WGBS`, `DNase-seq`, `ATAC-seq`, `PRO-cap`, `PRO-seq`, `Bru-seq`, `BruChase-seq`, `RAMPAGE`, `PAS-seq` and those that had available bigWig files for GrCh38. We got over 100 `eCLIP` annotations of RNA binding; since each of these annotations are sparse, we summed them together to create a single `all_eCLIP` annotation. For `TF ChIP-seq` experiments that targeted a transcription factor, we only used assays from the 24 targets that had measurements from two or more labs.

Each of these experiments had multiple data tracks. We used the `fold_change_over_control` for a random replicate if it was available, otherwise we used a randomly chosen track. In total we had 111 tracks from ENCODE; the full list with bioproject ids is as follows: `all_eCLIP`, `TF_ChIP-seq of MTA3 (ENCSR391KQC)`, `TF_ChIP-seq of MCM3 (ENCSR990AZC)`, `TF_ChIP-seq of POLR2AphosphoS5 (ENCSR000BTW)`, `Histone_ChIP-seq of H3K27ac (ENCSR601VHO)`, `TF_ChIP-seq of NFIB (ENCSR702BYX)`, `ChIA-PET of CTCF (ENCSR514HBO)`, `TF_ChIP-seq of SUZ12 (ENCSR757EMK)`, `Histone_ChIP-seq of H3K9me3 (ENCSR999HNE)`, `TF_ChIP-seq of CAMTA2 (ENCSR336GFK)`, `ChIA-PET of POLR2A (ENCSR447IUA)`, `TF_ChIP-seq of NFRKB (ENCSR145BHD)`, `TF_ChIP-seq of SIN3A (ENCSR468LUO)`, `Histone_ChIP-seq of H3K27me3 (ENCSR197KBA)`, `PAS-seq (ENCSR055TUB)`, `Bru-seq (ENCSR258ARX)`, `polyA_minus_RNA-seq (ENCSR000CQI)`, `TF_ChIP-seq of HLTF (ENCSR090JNM)`, `TF_ChIP-seq`

of FOXK2 (ENCSR465VLK), TF_ChIP-seq of CBX8 (ENCSR616MOB), TF_ChIP-seq of ZFX (ENCSR503GVO), ATAC-seq (ENCSR890DWH), TF_ChIP-seq of TARDBP (ENCSR412QBS), DNase-seq (ENCSR367FKP), Histone_ChIP-seq of H3K4me1 (ENCSR238WIK), TF_ChIP-seq of GATAD2A (ENCSR160QYK), TF_ChIP-seq of ARNT (ENCSR613NUC), TF_ChIP-seq of PKNOX1 (ENCSR115SMW), TF_ChIP-seq of MCM7 (ENCSR542WJU), TF_ChIP-seq of MLLT1 (ENCSR427BBI), RAMPAGE (ENCSR413FKS), TF_ChIP-seq of HDAC1 (ENCSR711VWL), Histone_ChIP-seq of H3K27ac (ENCSR400XSW), TF_ChIP-seq of Cebpa (ENCSR334SSD), TF_ChIP-seq of DPF2 (ENCSR715CCR), Histone_ChIP-seq of H3K4me2 (ENCSR693KAX), PAS-seq (ENCSR014VJO), Histone_ChIP-seq of H2AFZ (ENCSR859FGW), TF_ChIP-seq of CTBP1 (ENCSR636EYA), TF_ChIP-seq of SMARCA5 (ENCSR895HSJ), polyA_minus_RNA-seq (ENCSR000CQH), Histone_ChIP-seq of H4K20me1 (ENCSR839YFS), TF_ChIP-seq of BCOR (ENCSR808AKZ), TF_ChIP-seq of GTF2F1 (ENCSR557JTZ), Histone_ChIP-seq of H3K56ac (ENCSR036NSK), BruChase-seq (ENCSR245MXB), TF_ChIP-seq of CTCF (ENCSR035OXA), TF_ChIP-seq of JUNB (ENCSR431LRW), TF_ChIP-seq of TRIM24 (ENCSR957LDM), TF_ChIP-seq of NBN (ENCSR278SQL), Histone_ChIP-seq of H3K27me3 (ENCSR374JBS), Histone_ChIP-seq of H3K36me3 (ENCSR845BEG), TF_ChIP-seq of MAX (ENCSR000BTY), TF_ChIP-seq of LARP7 (ENCSR725ELR), microRNA-seq (ENCSR496QLS), Histone_ChIP-seq of H3K9me3 (ENCSR623YMO), TF_ChIP-seq of JUN (ENCSR192PBJ), TF_ChIP-seq of PLRG1 (ENCSR019KPC), TF_ChIP-seq of MLX (ENCSR125DAD), Histone_ChIP-seq of H3K9ac (ENCSR705USK), BruChase-seq (ENCSR809IJG), TF_ChIP-seq of GATAD2B (ENCSR389BLX), TF_ChIP-seq of KHSRP (ENCSR686EYO), ChIA-PET of POLR2A (ENCSR982KEM), Histone_ChIP-seq of H2AFZ (ENCSR256KRN), Histone_ChIP-seq of H3K4me1 (ENCSR716KBL), TF_ChIP-seq of SMARCA4 (ENCSR587OQL), TF_ChIP-seq of PHB (ENCSR650AWW), Histone_ChIP-seq of H3K36me3 (ENCSR472CMR), RAMPAGE (ENCSR773EZK), ATAC-seq (ENCSR677MJF), Histone_ChIP-seq of H4K5ac (ENCSR035BZI), DNase-seq (ENCSR255STJ), WGBS (ENCSR166VVF), PRO-cap (ENCSR935RNW), TF_ChIP-seq of CSDE1 (ENCSR626QJQ), TF_ChIP-seq of DMAP1 (ENCSR670YPQ), Histone_ChIP-seq of H3K4me2 (ENCSR714QUE), TF_ChIP-seq of PBX3 (ENCSR000BVE), TF_ChIP-seq of HDGF (ENCSR563YDA), Histone_ChIP-seq of H3K23ac (ENCSR473AQI), TF_ChIP-seq of CEBPB (ENCSR000BUB), TF_ChIP-seq of RFXANK (ENCSR823ADL), small_RNA-seq (ENCSR000CSZ), PRO-seq (ENCSR989CPK), Bru-seq (ENCSR892NYB), TF_ChIP-seq of MNT (ENCSR730TBC), TF_ChIP-seq of RBBP5 (ENCSR330EXS), TF_ChIP-seq of NONO (ENCSR912NMR), summed_RNA_binding of e (clip), TF_ChIP-seq of POLR2A (ENCSR388QZF), TF_ChIP-seq of CBFA2T3 (ENCSR697YLJ), TF_ChIP-seq of YBX3 (ENCSR567JEU), TF_ChIP-seq of RAD21 (ENCSR000BUC), TF_ChIP-seq of JUND (ENCSR000BSK), Histone_ChIP-seq of H3K79me1 (ENCSR213JMO), TF_ChIP-seq of MTA2 (ENCSR411UYA), small_RNA-seq (ENCSR000CSF), TF_ChIP-seq of FOXP1 (ENCSR369YUK), TF_ChIP-seq of IKZF1 (ENCSR278JQG), TF_ChIP-seq of ZBTB1 (ENCSR309ELI), TF_ChIP-seq of NCOA3 (ENCSR573OJP), TF_ChIP-seq of CREB1 (ENCSR620DUQ), Histone_ChIP-seq of H2BK20ac (ENCSR462XRE), PRO-cap (ENCSR098LLB), TF_ChIP-seq of SUPT5H (ENCSR894CGX), TF_ChIP-seq of EP300 (ENCSR686BQM), TF_ChIP-seq of SP1 (ENCSR334KIQ), Histone_ChIP-seq of H3K4me3 (ENCSR105FGG), TF_ChIP-seq of HDAC2 (ENCSR659LJJ), TF_ChIP-seq of NR3C1 (ENCSR355HLV).