Supplementary Infomation

A Supplementary Figures

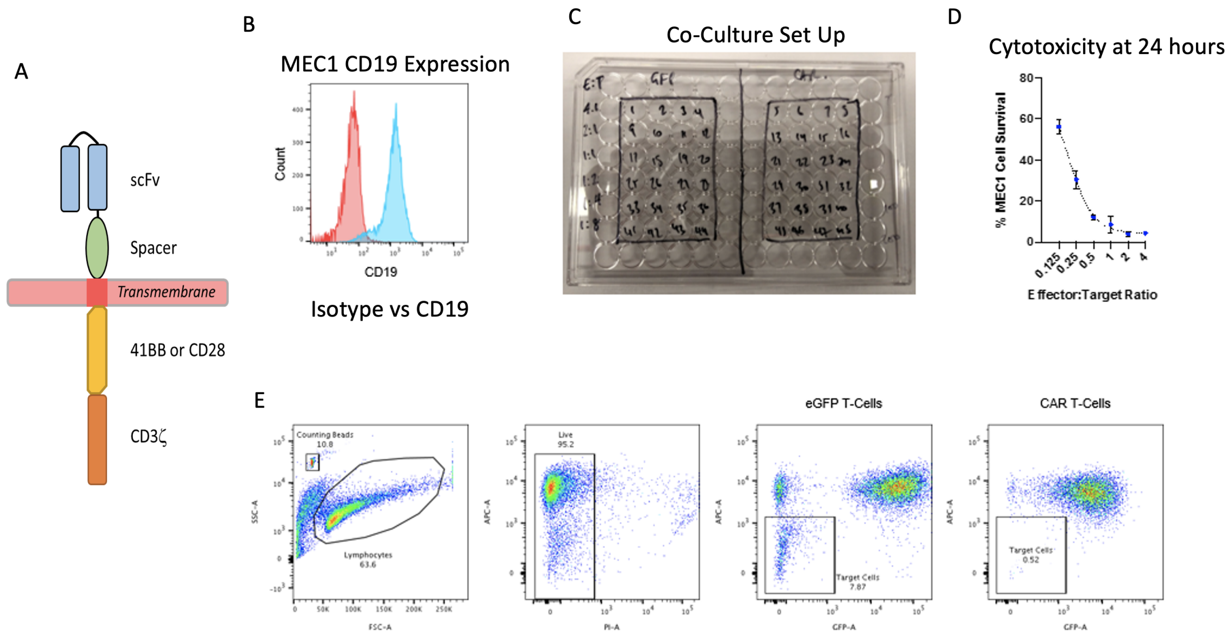

Figure S1: **Experimental details for CAR-T. (A)** Structure of CAR-T protein. Experiments A,B, and C used the CD28 protein while experiment D used 41BB. (**B**) Flow cytometry plot showing CD19 staining (blue) versus isotype control (red) in MEC1 cells. (**C**) Co-culture set up in 96 well plate. (**D**) Demonstrated dose-response cytotoxicity for different ratios of effector:target cells. **E.** Representative gating strategy for isolating CAR-T cells for use in experiment

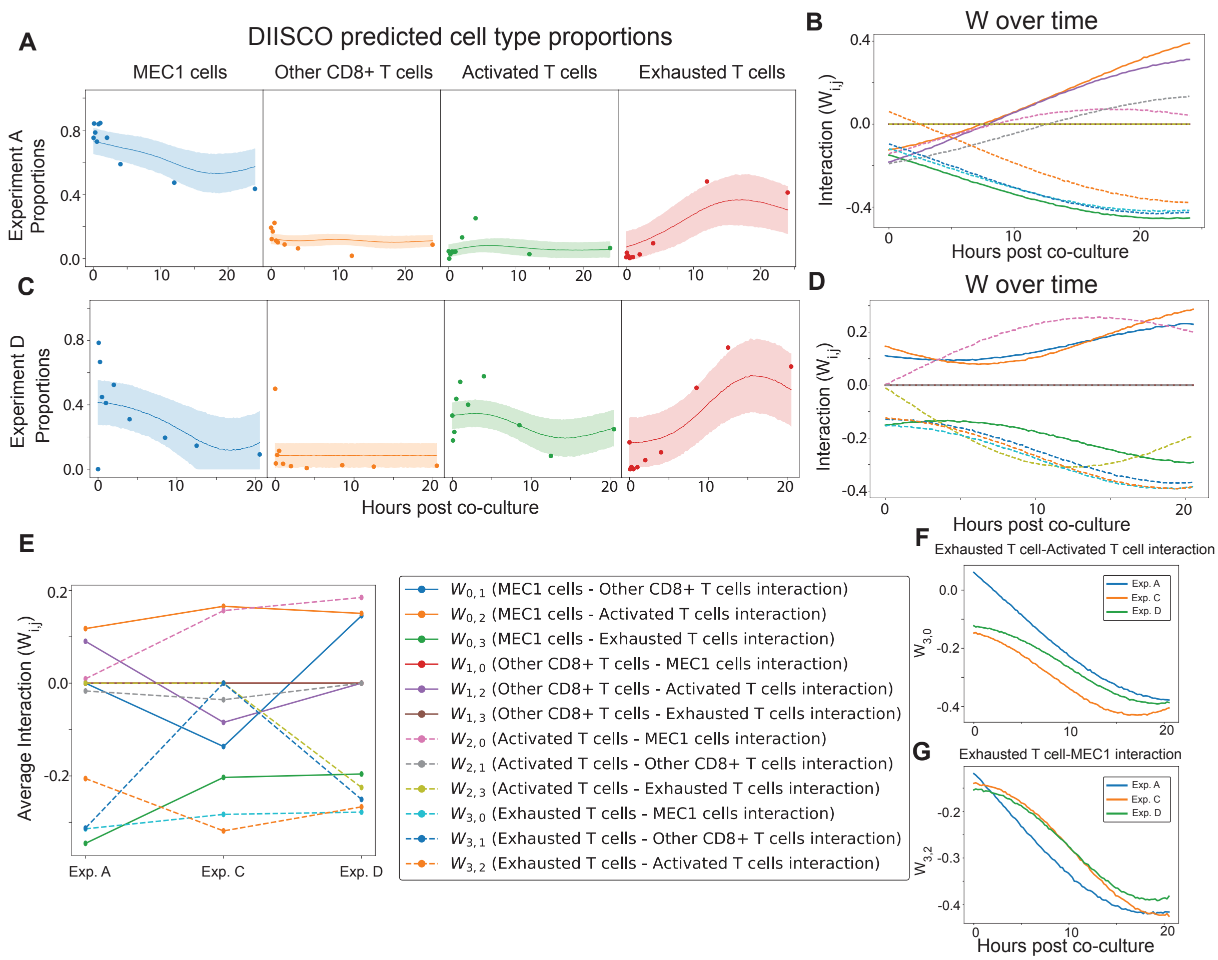

Figure S2: **DIISCO performance on additional replicate experiments (A)**. Learned proportions from DIISCO for experiment A. Dots represent calculated proportions at each time point, line represents mean prediction and shaded region depicts 85% percentile confidence region. (**B**) Learned W over time for experiment A. (**C**) Inferred proportions from DIISCO for experiment D. (**D**) Learned W over time for experiment D. (**E-F**) W dynamics over time for interactions between Exhausted-Activated T cells (E) and Exhausted MEC1 cells (F) across experiments A, C, D. (**G**) Average W interaction score across all experiments.

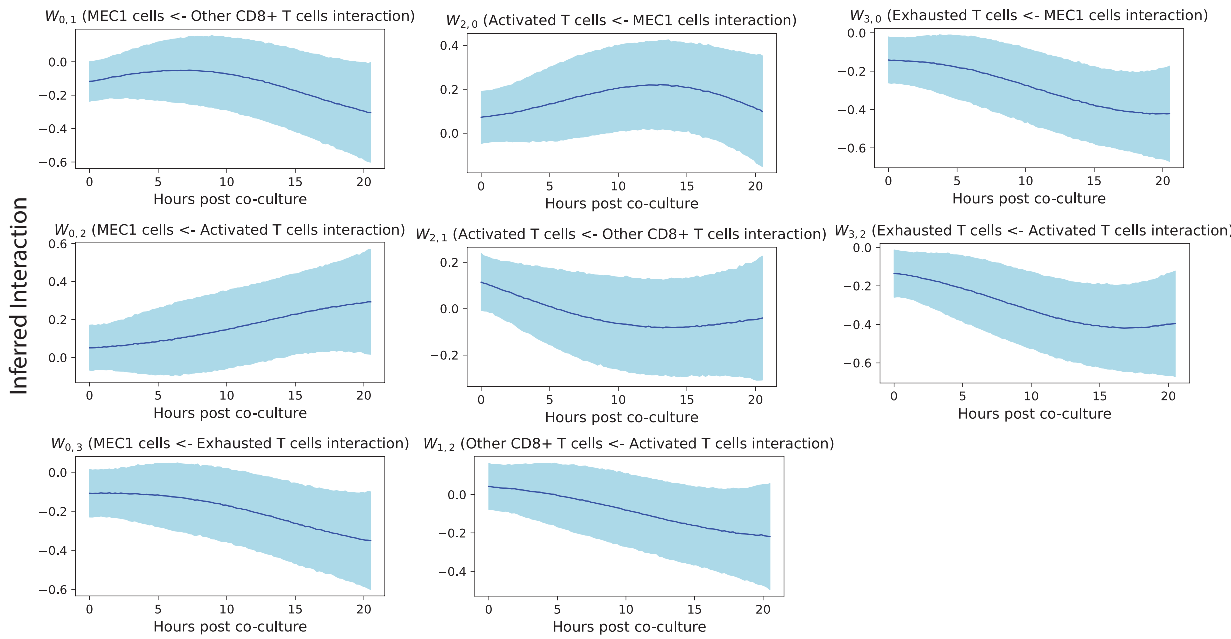

Figure S3: **Confidence intervals for W predictions in Experiment C.** All non-zero interactions are shown, blue line depicts mean predicted interaction over time while the shaded region depicts the 85% confidence interval.

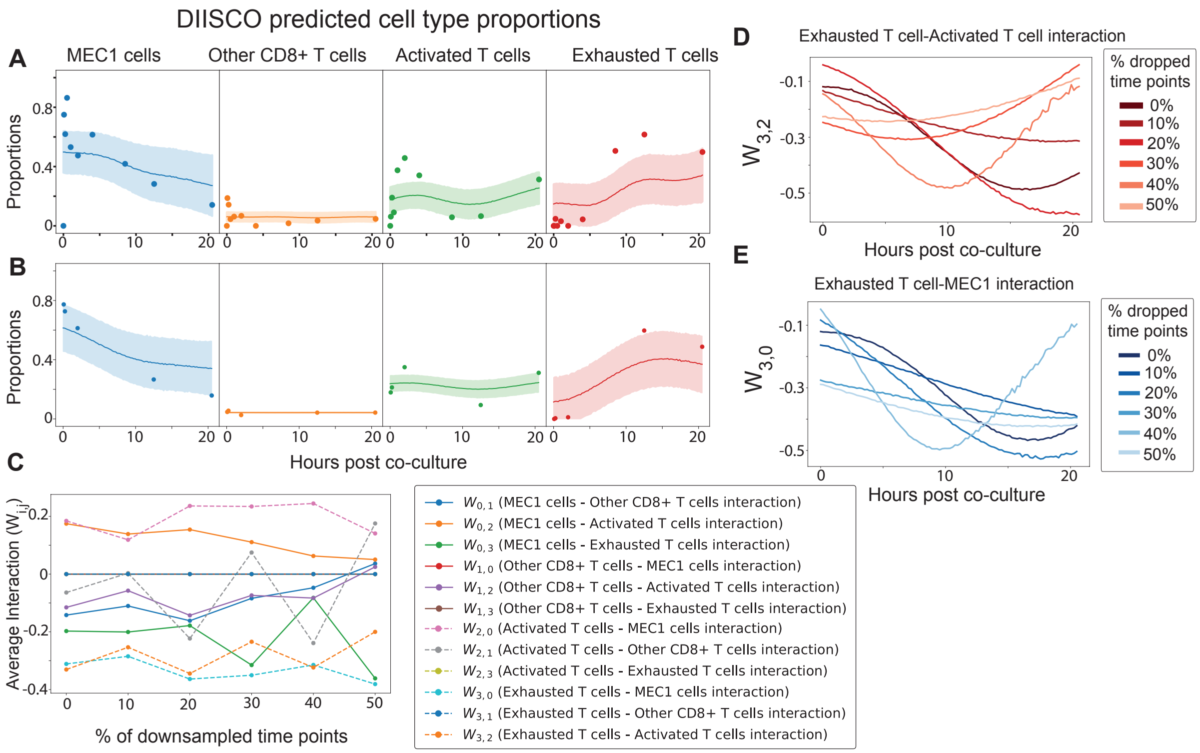

Figure S4: **DIISCO robustness to downsampling.** (**A**) DIISCO predicted cell type proportions when downsampling and removing 90% of cells from the data. (**B**) DIISCO predicted cell type proportions when downsampling and removing 50% of time points. (**C**) Average W inferred interaction for varying numbers of time points. (**D**) Exhausted-Activated T cell interaction over time for varying downsampled time points. (**E**) Exhausted T cell - MEC1 interaction over time for varying downsampled time points.

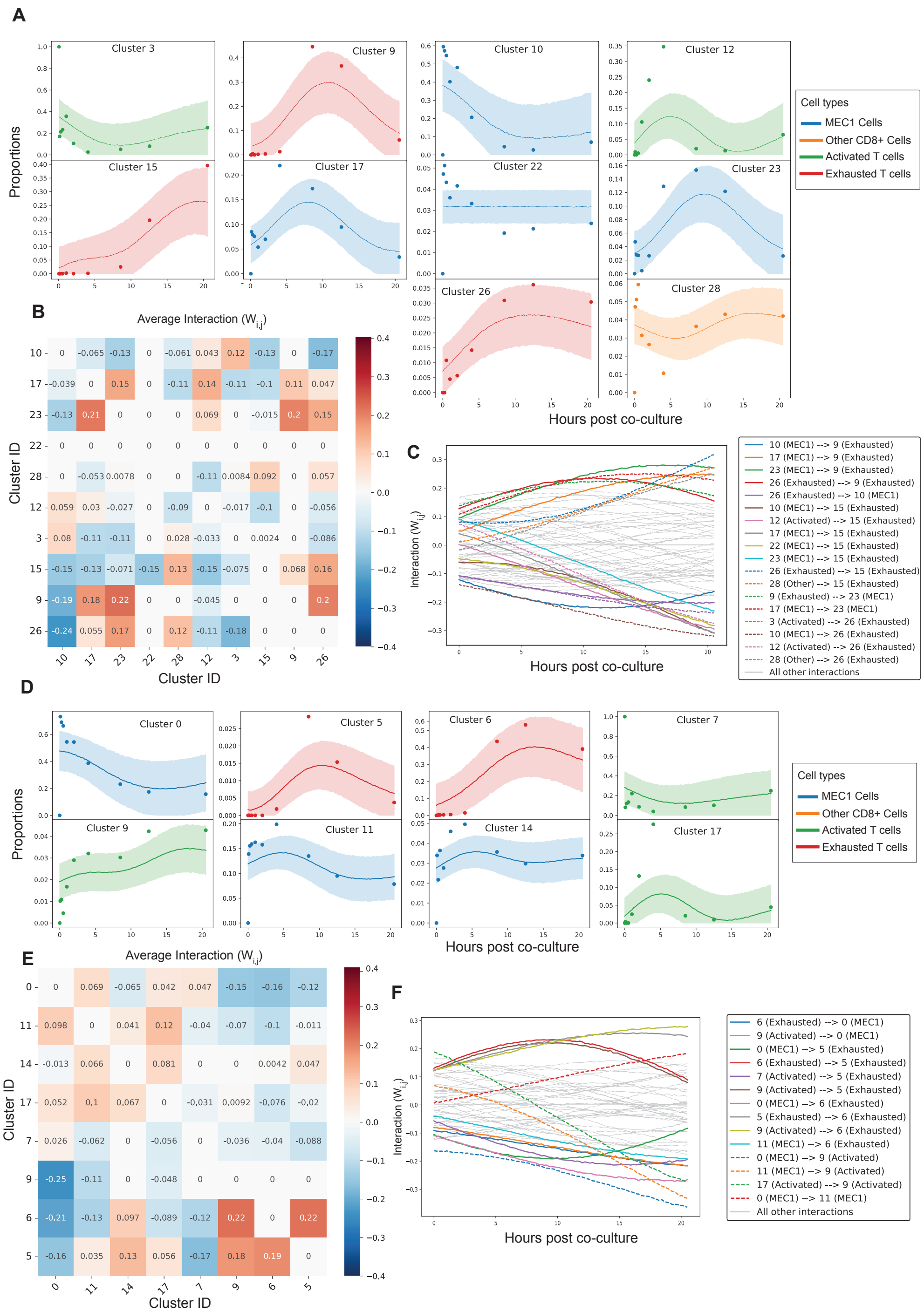

Figure S5: **DIISCO robustness to clustering method.** (**A**) DIISCO predicted cell type dynamics on individual Phenograph clusters (without grouping into metaclusters). Cells colored by metacluster cell type assignment. (**B**) Average interaction between all cluster pairs. (**C**) Interaction over time. (**D**) Predicted cell type dynamics when applying DIISCO to individual Leiden clusters. Cells colored by cell type assignment. (**E**) Average interaction between all cluster pairs. (**F**) Interaction over time.

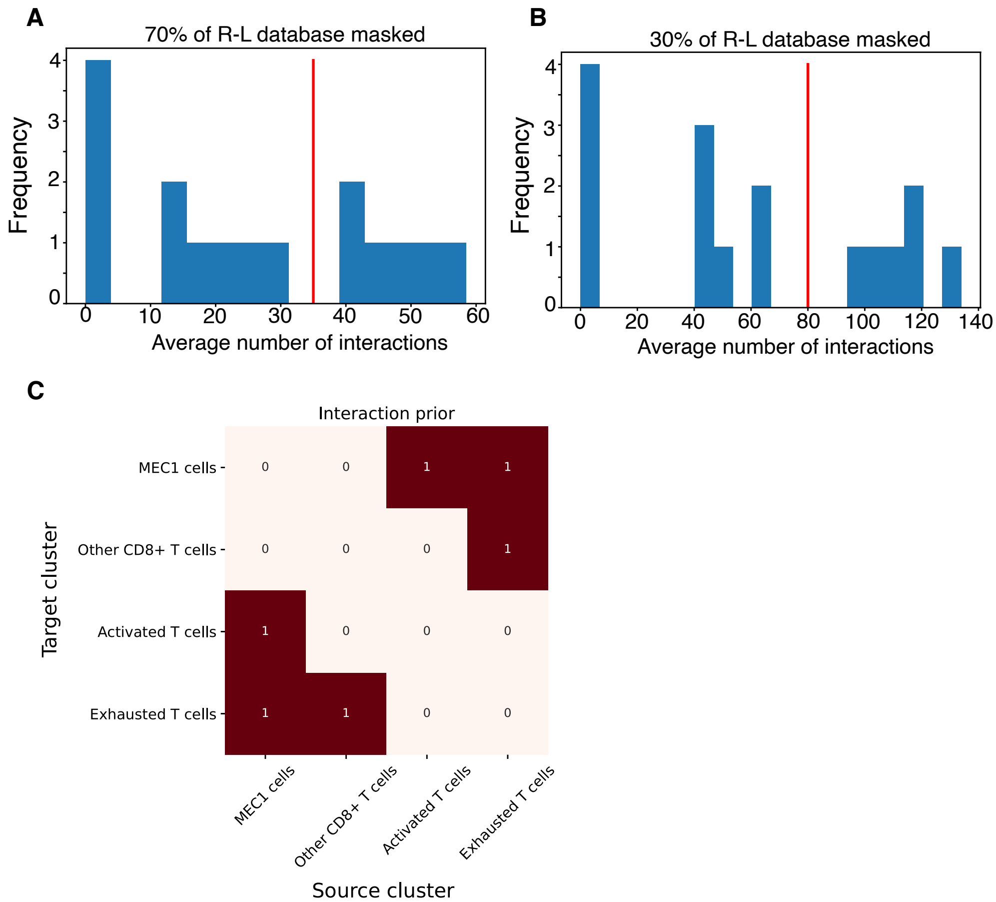

Figure S6: **Adjusting binarization threshold to compensate for incomplete R-L databases.** **(A)** Average predicted interactions from OmnipathDB when 30% of R-L interactions are masked from database. Red line indicates binarization threshold. (**B**) Average predicted interactions from OmnipathDB when 70% of R-L interactions are masked from database. Red line indicates binarization threshold. (**C**) Prior matrix used in DIISCO model, generated based on user defined thresholds. Both **A** and **B** threshold choices, as indicated by the red lines, lead to the same interaction prior matrix.

Model	# timepoints	R2_Y	RMSE_Y	AUC	AUPRC	F1
DIISCO	10	**0.999±0.0**	0.03±0.0	**0.975±0.02**	**0.922±0.08**	0.855±0.03
LM_PRIOR	10	0.687±0.17	0.577±0.26	0.965±0.03	0.888±0.08	**0.908±0.05**
LM	10	**1.0±0.0**	**0.0±0.0**	0.572±0.01	0.304±0.01	0.347±0.01
RLM_PRIOR	10	0.803±0.13	0.439±0.15	**0.981±0.01**	**0.904±0.07**	0.87±0.05
RLM	10	0.951±0.03	0.211±0.09	0.631±0.02	0.445±0.03	0.445±0.03
DIISCO	20	**0.999±0.0**	0.034±0.0	**0.977±0.01**	**0.931±0.03**	0.854±0.01
LM_PRIOR	20	0.601±0.08	0.635±0.13	0.951±0.03	0.872±0.06	**0.888±0.03**
LM	20	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.305±0.0	0.346±0.0
RLM_PRIOR	20	0.903±0.05	0.302±0.08	**0.983±0.01**	**0.923±0.03**	**0.886±0.02**
RLM	20	0.976±0.01	0.148±0.04	0.613±0.03	0.421±0.03	0.437±0.02
DIISCO	30	**0.999±0.0**	0.039±0.0	**0.982±0.01**	**0.946±0.02**	0.871±0.02
LM_PRIOR	30	0.54±0.04	0.726±0.06	0.964±0.02	0.9±0.04	**0.9±0.01**
LM	30	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.304±0.0	0.346±0.0
RLM_PRIOR	30	0.928±0.05	0.273±0.1	**0.983±0.0**	0.917±0.03	**0.896±0.02**
RLM	30	0.989±0.01	0.11±0.02	0.635±0.02	0.417±0.03	0.442±0.01
DIISCO	40	**0.998±0.0**	0.041±0.0	**0.982±0.0**	**0.952±0.01**	0.859±0.01
LM_PRIOR	40	0.54±0.05	0.726±0.07	0.973±0.01	0.917±0.03	**0.906±0.01**
LM	40	**1.0±0.0**	**0.0±0.0**	0.569±0.0	0.306±0.0	0.344±0.0
RLM_PRIOR	40	0.946±0.03	0.236±0.06	**0.983±0.01**	**0.913±0.04**	**0.9±0.01**
RLM	40	0.989±0.01	0.107±0.02	0.617±0.02	0.4±0.02	0.433±0.01
DIISCO	60	**0.998±0.0**	0.042±0.0	0.981±0.0	**0.947±0.01**	0.859±0.01
LM_PRIOR	60	0.517±0.03	0.748±0.06	0.949±0.03	0.88±0.03	**0.88±0.03**
LM	60	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.305±0.0	0.346±0.0
RLM_PRIOR	60	0.972±0.01	0.176±0.02	**0.983±0.0**	0.918±0.02	**0.892±0.01**
RLM	60	0.993±0.0	0.087±0.01	0.627±0.02	0.409±0.02	0.437±0.01
DIISCO	70	**0.998±0.0**	0.043±0.0	0.981±0.0	**0.946±0.02**	0.863±0.02
LM_PRIOR	70	0.545±0.03	0.705±0.05	0.968±0.02	0.915±0.03	**0.894±0.02**
LM	70	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.304±0.0	0.346±0.0
RLM_PRIOR	70	0.98±0.0	0.149±0.01	**0.983±0.0**	0.915±0.01	**0.889±0.01**
RLM	70	0.994±0.0	0.08±0.0	0.627±0.01	0.414±0.01	0.436±0.01

Table S1: **Method performance for varying number of timepoints**. Noise parameter for dynamics set by ϵ, which is a random variable sampled from a normal distribution with standard deviation of 0.1, as described in **Methods**. R^2 calculated between inferred and ground-truth $W(t)$. Mean and SD across 10 iterations are shown. Model acronyms denote the following: LM-PRIOR = Linear Model with prior. LM = Linear Model. RLM-PRIOR = Rolling Linear Model with prior. RLM = Rolling Linear Model. Model details can be found in **Methods**. Comparison metrics used are as follows: R^2_Y, R^2_W: R^2 value comparing predictions to ground truth for dynamics (Y) or interactions (W). Higher is better. $RMSE\_Y$, $RMSE\_W$: Root mean squared error for dynamics (Y) or interactions (W). Lower is better. AUC: Area under ROC curve. Higher is better. AUPRC: Area under Precision-Recall curve. Higher is better. F1: Max F1 score. Higher is better. AUC, AUPRC, and F1 scores calculated comparing predicted interactions to ground truth interactions.

Model	# timepoints	R2_Y	RMSE_Y	AUC	AUPRC	F1
DIISCO	10	**0.999±0.0**	0.039±0.01	0.936±0.01	0.762±0.05	0.849±0.02
LM_PRIOR	10	0.671±0.13	0.734±0.23	0.958±0.02	0.846±0.08	**0.907±0.02**
LM	10	**1.0±0.0**	**0.0±0.0**	0.569±0.01	0.306±0.01	0.344±0.01
RLM_PRIOR	10	0.374±0.55	0.954±0.45	**0.982±0.01**	**0.924±0.05**	0.897±0.04
RLM	10	0.848±0.12	0.481±0.19	0.597±0.04	0.397±0.05	0.44±0.02
DIISCO	20	**0.999±0.0**	0.045±0.0	0.938±0.01	0.755±0.05	0.847±0.01
LM_PRIOR	20	0.506±0.07	0.886±0.09	0.962±0.01	0.876±0.06	**0.905±0.01**
LM	20	**1.0±0.0**	**0.0±0.0**	0.57±0.01	0.306±0.0	0.344±0.01
RLM_PRIOR	20	-0.002±0.84	1.163±0.39	**0.978±0.0**	**0.896±0.04**	0.863±0.02
RLM	20	0.895±0.04	0.404±0.08	0.592±0.03	0.377±0.04	0.427±0.01
DIISCO	30	**0.999±0.0**	0.046±0.0	0.947±0.01	0.804±0.06	0.843±0.01
LM_PRIOR	30	0.517±0.05	0.874±0.1	0.963±0.01	0.885±0.03	**0.901±0.01**
LM	30	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.304±0.0	0.346±0.0
RLM_PRIOR	30	0.275±0.75	0.988±0.44	**0.98±0.0**	**0.902±0.02**	0.87±0.02
RLM	30	0.889±0.03	0.413±0.05	0.6±0.02	0.373±0.03	0.427±0.01
DIISCO	40	**0.999±0.0**	0.048±0.0	0.953±0.01	0.831±0.03	0.845±0.0
LM_PRIOR	40	0.505±0.04	0.923±0.07	0.966±0.01	**0.895±0.03**	**0.903±0.01**
LM	40	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.305±0.0	0.345±0.0
RLM_PRIOR	40	-0.246±1.08	1.327±0.66	**0.979±0.0**	**0.895±0.03**	0.869±0.02
RLM	40	0.918±0.02	0.372±0.03	0.597±0.02	0.362±0.02	0.426±0.01
DIISCO	60	**0.999±0.0**	0.049±0.0	0.947±0.01	0.802±0.04	0.842±0.0
LM_PRIOR	60	0.502±0.03	0.91±0.05	0.961±0.01	0.878±0.02	**0.9±0.0**
LM	60	**1.0±0.0**	**0.0±0.0**	0.571±0.0	0.304±0.0	0.346±0.0
RLM_PRIOR	60	0.516±0.16	0.886±0.15	**0.98±0.0**	**0.904±0.01**	0.859±0.01
RLM	60	0.915±0.01	0.374±0.03	0.595±0.01	0.367±0.01	0.419±0.01
DIISCO	70	**0.998±0.0**	0.049±0.0	0.943±0.01	0.79±0.05	0.841±0.0
LM_PRIOR	70	0.501±0.03	0.896±0.07	0.945±0.03	0.849±0.04	**0.886±0.03**
LM	70	**1.0±0.0**	**0.0±0.0**	0.572±0.0	0.304±0.0	0.347±0.0
RLM_PRIOR	70	0.438±0.21	0.935±0.17	**0.978±0.0**	**0.893±0.01**	0.863±0.01
RLM	70	0.912±0.01	0.375±0.02	0.591±0.01	0.365±0.01	0.417±0.0

Table S2: **Method performance for varying number of timepoints on noisier dynamics**. Noise parameter for dynamics set by ϵ, which is a random variable sampled from a normal distribution with standard deviation of 0.5, as described in **Methods**. R^2 calculated between inferred and ground-truth $W(t)$. Mean and SD across 10 iterations are shown. Model acronyms denote the following: LM-PRIOR = Linear Model with prior. LM = Linear Model. RLM-PRIOR = Rolling Linear Model with prior. RLM = Rolling Linear Model. Model details can be found in **Methods**. Comparison metrics used are as follows: R^2_Y, R^2_W: R^2 value comparing predictions to ground truth for dynamics (Y) or interactions (W). Higher is better. $RMSE\_Y$, $RMSE\_W$: Root mean squared error for dynamics (Y) or interactions (W). Lower is better. AUC: Area under ROC curve. Higher is better. AUPRC: Area under Precision-Recall curve. Higher is better. F1: Max F1 score. Higher is better. AUC, AUPRC, and F1 scores calculated comparing predicted interactions to ground truth interactions.

Experiment	CAR	Effector:Target Ratio	Time post co-culture	Hashing antibody 1	Hashing antibody 2	Counts
A	28z	1:1	24h	1	2	190
	28z	1:1	12h	1	3	112
	28z	1:1	4h	1	4	263
	28z	1:1	2h	1	5	390
	28z	1:1	1h	2	3	756
	28z	1:1	45min	2	4	375
	28z	1:1	30min	2	5	580
	28z	1:1	15min	3	4	704
	28z	1:1	5min	3	5	872
	28z	1:1	0min	4	5	7756
C	28z	1:1	20.5h	1	2	3
	28z	1:1	12.5h	1	3	289
	28z	1:1	8.5h	1	4	545
	28z	1:1	4h	1	5	431
	28z	1:1	2h	2	3	994
	28z	1:1	1h	2	4	1065
	28z	1:1	30min	2	5	1248
	28z	1:1	15min	3	4	2591
	28z	1:1	5min	3	5	2454
	28z	1:1	0min	4	5	4219
D	41BBz	1:1	20.5h	1	2	13
	41BBz	1:1	12.5h	1	3	120
	41BBz	1:1	8.5h	1	4	411
	41BBz	1:1	4h	1	5	609
	41BBz	1:1	2h	2	3	790
	41BBz	1:1	1h	2	4	468
	41BBz	1:1	30min	2	5	1200
	41BBz	1:1	15min	3	4	2774
	41BBz	1:1	5min	3	5	1890
	41BBz	1:1	0min	4	5	2924

Table S3: **Experimental details for each co-culture experiment.**

B Justification of Inference Algorithm

According to the model, we are interested in computing the posterior

$$p(\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u, \mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o).$$

Although, it is not possible to tractably compute or sample from this distribution, we can use its structure to obtain a reasonable approximation. First, using the chain rule of probability, we have:

$$\begin{aligned} p(\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u, \mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o) = \; & p(\mathcal{Y}_u \mid \mathcal{W}_u, \mathcal{F}_u, \mathcal{W}_o, \mathcal{F}_o, \mathcal{Y}_o) && (8)\\ & p(\mathcal{W}_u \mid \mathcal{F}_u, \mathcal{W}_o, \mathcal{F}_o, \mathcal{Y}_o) && (9)\\ & p(\mathcal{F}_u \mid \mathcal{W}_o, \mathcal{F}_o, \mathcal{Y}_o) && (10)\\ & p(\mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o). && (11) \end{aligned}$$

However, based on Figure (1) we see that in this factorization some dependencies are irrelevant. In particular, we note that the observations $\mathcal{Y}_o$ are independent of everything else given $\mathcal{W}_o$ and $\mathcal{F}_o$. Therefore, equation (8) can be written as $p(\mathcal{Y}_u \mid \mathcal{W}_u, \mathcal{F}_u)$, that conditioned on $\mathcal{W}_o$, $\mathcal{W}_u$ is independent of everything. Hence, equation (9) can be written as $p(\mathcal{W}_u \mid \mathcal{W}_o)$, and a similar relationship holds between $\mathcal{F}_u$ and $\mathcal{F}_o$, so equation (10) can be written as $p(\mathcal{F}_u \mid \mathcal{F}_o)$.

Using these simplifications, we have:

$$p(\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u, \mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o) = \; p(\mathcal{Y}_u \mid \mathcal{W}_u, \mathcal{F}_u)\; p(\mathcal{W}_u \mid \mathcal{W}_o)\; p(\mathcal{F}_u \mid \mathcal{F}_o)\; p(\mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o). \qquad (12)$$

Consequently, if we can obtain a good approximation to the last term, and the first three terms on the right hand side are tractable to compute, we can obtain a good approximation to the full posterior by performing ancestral sampling where we first sample from our approximation $p(\mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o)$ and then condition $p(\mathcal{Y}_u \mid \mathcal{W}_u, \mathcal{F}_u)$, $p(\mathcal{W}_u \mid \mathcal{W}_o)$, and $p(\mathcal{F}_u \mid \mathcal{F}_o)$. In the next sections, we describe how we obtain an approximation to to $p(\mathcal{W}_o, \mathcal{F}_o \mid \mathcal{Y}_o)$ and provide a brief description of how we perform ancestral sampling.

C Inference Algorithm Details

The simplified inference algorithm is shown in Algorithm 2.

C.1 Ancestral Sampling.

To perform ancestral sampling, we execute the following steps:

1. Sample $\mathcal{W}_o$ and $\mathcal{F}_o$ from $q_\phi(\mathcal{W}_o, \mathcal{F}_o)$.
2. Compute the posterior distribution $p(\mathcal{W}_u \mid \mathcal{W}_o)$ using the samples from step 1 using algorithm 2.1 from (Rasmussen and I., 2008) and sample $\mathcal{W}_u$ from it.
3. Compute the posterior distribution $p(\mathcal{F}_u \mid \mathcal{F}_o)$ using the samples from step 1 using algorithm 2.1 from (Rasmussen and I., 2008) and sample $\mathcal{F}_u$ from it.
4. Compute the posterior distribution $p(\mathcal{Y}_u \mid \mathcal{W}_u, \mathcal{F}_u)$ using equation (1) and sample $\mathcal{Y}_u$ from it.
5. Return $\mathcal{Y}_u$, $\mathcal{W}_u$, $\mathcal{F}_u$, $\mathcal{W}_o$, and $\mathcal{F}_o$.

In practice, since steps 2 and 3 are computationally expensive due to the computation of the posterior of a Gaussian Process, we sample $p(\mathcal{W}_u \mid \mathcal{W}_o)$ and $p(\mathcal{F}_u \mid \mathcal{F}_o)$ multiple times per sampling of $\mathcal{W}_o$ and $\mathcal{F}_o$ respectively.

C.1.1 Additional Practical Considerations.

During training, we use early stopping by defining an epoch as 1000 iterations of the optimization algorithm and stopping when the ELBO has not increased for 10 epochs. For hyper-parameter selection, we follow the recommendations detailed in Supplementary Information Section D but set a hyper-prior on the length scale of $W(t)$ to allow for flexibility in the model. To infer this value we augment the variational family above with an additional term $q_{\phi_{\tau_w}}(\tau_w) = \delta(\exp(\phi_{\tau_w}))$ where δ is the delta distribution. As further discussed in Supplementary Information Section D we emphasize that choosing these hyper-parameters is crucial for the model to adequately perform its function as incorrectly setting these values can lead to degenerate solutions with non-identifiability.

Algorithm 2 Simplified Inference Algorithm used by DIISCO

Input: Set of time points $\mathcal{T}$, Number of latent features K, Noise covariance σ_y^2.
Initialize ϕ.
while not converged **do**
  for $i \in [N_{\text{elbo}}]$ **do**
    $\epsilon_i \sim \mathcal{D}$
  end for
  $\phi \leftarrow \phi - \alpha \frac{1}{N_{\text{elbo}}} \sum_{i=1}^{N_{\text{elbo}}} \nabla_\phi h_\phi(z(\epsilon_i, \phi))$
end while

for $s \in \{1, \ldots, N_{\text{samples}}\}$ **do**
  $(\mathcal{W}_o^s, \mathcal{F}_o^s) \sim q_\phi(\mathcal{W}_o, \mathcal{F}_o)$
  $\mathcal{Y}_u^s \sim p(\mathcal{Y}_u | \mathcal{W}_u^s, \mathcal{F}_u^s)$
  $\mathcal{W}_u^s \sim p(\mathcal{W}_u | \mathcal{W}_o^s)$ ▷ Using Algorithm 2.1 in (Rasmussen and I., 2008)
  $\mathcal{F}_u^s \sim p(\mathcal{F}_u | \mathcal{F}_o^s)$ ▷ Using Algorithm 2.1 (Rasmussen and I., 2008)
  $\mathcal{Y}_u^s \sim p(\mathcal{Y}_u | \mathcal{W}_u^s, \mathcal{F}_u^s)$
end for

Return $\{(\mathcal{W}_o^s, \mathcal{F}_o^s, \mathcal{W}_u^s, \mathcal{W}_u^s, \mathcal{Y}_u^s)\}_{s \in [N_{\text{samples}}]}$

D Hyper-parameter Selection Guide

Choosing the adequate hyper-parameters is crucial for the success of the model. In particular, a suboptimal selection of hyper-parameters can lead to non-identifiability.

In this section, we provide a summary of the most relevant hyper-parameters of the model, their interpretation, and suggestions and reasoning for how to set them.

Table S4: Hyperparameters and their Descriptions

Symbol	Description
τ_f	Lengthscale for f, controls how flexible is the prior over the latent features.
τ_w	Lengthscale for W, controls how flexible is the matrix and how much information is shared across time points.
v_f	Variance for f, controls the magnitude of the latent features.
v_w	Variance for W, controls the magnitude of W matrix
σ_f	Standard deviation for f, controls the amount of non informative noise we believe is in the latent features.
σ_w	Standard deviation for W, controls the amount of non informative noise we believe is in W and serves mostly as a stability parameter during optimization.
σ_y	Standard deviation for y, controls the amount of noise the model assumes is in the data.

Table S4 contains a summary of the hyper-parameters used by the model. We will describe below the role that each of these hyper-parameters play and how to set them. We will go from most important one to least important one:

1. Lengthscale τ_w: This is by far the most important parameter of the model. It controls the flexibility and smoothness of the W matrix which determines both how much information is used to inform the value at a predicted time point and how quickly this value changes. Ideally, one would like to set it from domain knowledge but it can be set with intuition derived from the data as follows. Intuitively, two points a distance of a length-scale away have correlation of $\approx 0.6 \approx 1/2$ where this correlation is measured with respect to random function draws. A good rule of thumb is that the length scale should be roughly at least as big as the maximum distance in the data between the largest and the smallest of any j sequential data points, where j is the largest number of non-zero entries in a row of Λ. Mathematically,

$$\tau_w \geq \max\{|t_{i+j} - t_i| : i \in \{1 \ldots , t_{n-j}\}\}$$

where

$$j = \max\left\{\sum_{k'} \Lambda_{i,k'} : i \in 1, \ldots , K\right\}$$

Consequently, the sampling frequency of the data should be such that this length-scale is adequate to model the flexibility we expect in the W matrix. This is not a rule applicable everywhere but it adheres to the intuition that our model is performing a form of approximate local linear regression and this is the minimum number of points we would need for such a scheme to work approximately, taking into account the fact that a lot of information is being shared.

2. Lengthscale τ_f: This plays the normal role of the length-scale in traditional Gaussian Processes and can be set so that the prior matches the intuition of the user about the latent functions, or using one of the traditional methods implemented in any package for setting this hyper-parameter.

3. Variances v_f, v_w: These values play the same role as the variance in a standard Bayesian linear regression and can determine the magnitude of the functions drawn. In our case when

standardizing we set them to values slightly above one for f and higher for v_w to indicate a weak prior. As v_f affects the covariance, it should be handled jointly with τ_w to express beliefs about the flexibility of the functions.

4. Noise σ_f and σ_y: Indicate how much noise we believe is in our observations. The higher the noise, the more points the model will need to change the latent distribution. In our case, we used values lower than one to indicate low noise in our observations.

5. Noise σ_w: In our case, this is mostly an optimization stability parameter that can be set very small and is only useful to avoid numerical problems. We set it to 0.001 in our experiments and can be left to this default value.

E Assumptions for Application to Cell Type Proportions

Our model is designed to work equally with proportions as well as raw count data. However, one must have particular care when working with proportions to make sure that the assumptions of the model are met.

As detailed by Aitchison (Aitchison, 2003), compositional data, i.e data points that lie on the simplex, present a particular challenge when thinking about their correlation structure and what it implies about the real biological process.

In particular, if we assume that our proportions $y(t) \in \Delta^{K-1}$ emerge from some real process with some absolute number of counts $c(t) \in \mathbb{Z}^K_{\geq 0}$ and it is the case that $\sum_k c_k(t)$, i.e the total number of counts, varies widely through time then it is possible that the correlations that the model learns will not be meaningful. This is a limitation not only for this model but also for any model that only uses proportions to understand the relationship between the variables. If however, it is the case that $\sum_k c_k(t) \approx C$ for all t the inferences made by the model will be valid.

We demonstrate this issue with the following example. Assume that we have two clusters with the following dynamics:

$$c_1(t) = t^2 \tag{13}$$

$$c_2(t) = tc_1(t) \tag{14}$$

$$c_3(t) = t \tag{15}$$

And with proportions

$$y_1(t) = \frac{t^2}{t^3 + t^2 + t} \tag{16}$$

$$y_2(t) = \frac{t^3}{t^3 + t^2 + t} \tag{17}$$

$$y_3(t) = \frac{t}{t^3 + t^2 + t} \tag{18}$$

$$\tag{19}$$

Clearly $\sum_k c_k(t) = t^3 + t^2 + t$ is not constant. If one were working with raw values, we would like to say that c_1, c_2 positively interact in that they increase simultaneously. However, when considering proportions, the interpretation is different because now as y_2 is increasing, y_1 is decreasing which would suggest a negative interaction.

We thus advise using DIISCO in settings where the total number of cells across time points does not have extreme variability or to drop low-quality samples.

F Complexity: Further details

To determine the computational complexity of DIISCO we need to take into account the two steps in the algorithm: Approximate inference for estimating $P(\mathcal{W}_o, \mathcal{F}_o|\mathcal{Y}_o)$, and exact inference for $P(\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u|\mathcal{F}_o, \mathcal{W}_o)$. Below we describe the reasoning for the bounds provided in text for each of these steps.

F.1 Computing

We approximate $P(\mathcal{W}_o, \mathcal{F}_o, \mathcal{Y}_o)$ using stochastic variational inference (SVI). Each gradient step in SVI requires computing the estimate of the ELBO and backpropagating through it. In our case, this amounts to an estimate of the term.

$$\text{ELBO} = \mathbb{E}_{q_\phi(\mathcal{W}_o,\mathcal{F}_o)}[\log p_\theta(\mathcal{Y}_o,\mathcal{W}_o,\mathcal{F}_o) - \log q_\phi(\mathcal{W}_o,\mathcal{F}_o)] \tag{20}$$

where θ represents any hyper-parameters we might be simultaneously optimizing and ϕ represents the parameters of the variational family. In the algorithm, we obtain an estimate of this quantity by sampling from q, computing the log probability of the model using this sample (the first term in Eq 20), and computing the entropy of q analytically (the second term in Eq 20). Therefore, for computing the big O complexity we need to take three computations into account: 1) The complexity of computing the log probability 2) The complexity of computing the expectation of the q term and 3) The complexity of sampling. We describe these steps in detail for both variational families proposed in the paper.

As a reminder to the reader, the log probability allows the following factorization:

$$\left[\prod_k P(f_k(\mathcal{T}_o))\right]\left[\prod_{k,k'} P(W_{k,k'}(\mathcal{T}_o))\right]\left[\prod_{k,t} p(y_k(t)|\mathcal{F}_o,\mathcal{W}_o)\right]$$

where the terms $f_k(\mathcal{T}_o)$ and $W_{k,k'}(\mathcal{T}_o)$ denote that the coordinates f_k and $W_{k,k'}$ are evaluated at the time points in the set $\mathcal{T}_o$, and where the first two terms are made up of t dimensional GPs and the last term is a one-dimensional Gaussian distribution. We will use this factorization throughout.

Complexity of Fully Factorized Family

- **Computing the log probability**: First, we look at computing the log probability – the first term in the ELBO. The terms $P(f_k(\mathcal{T}_o)$ and $P(W_{k,k'}(\mathcal{T}_o))$ are Gaussian processes of one dimensions but with $|\mathcal{T}_o|$ timepoints. Computing this is $O(|\mathcal{T}_o|^3)$ because it requires inverting the covariance matrix. This can be cached so that we do it only once for all iterations but if we are computing the gradient with respect to the hyper-parameters as we do in our implementation, we have do it again for every iteration. We have to do this K times for $P(f_k(\mathcal{T}_o))$ and K^2 times for $P(W_{k,k'}(\mathcal{T}_o))$.

  To compute $p(y_k(t)|f(\mathcal{T}_o),W(\mathcal{T}_o))$ we need to multiply $W(t)f(t)$ for every t (which is $O(|\mathcal{T}_o|K^2)$) and have to compute the log probability which is in total $|\mathcal{T}_o|K\ O(1)$. Hence the complexity of the forward computation is

$$KO(|\mathcal{T}_o|^3) + K^2O(|\mathcal{T}_o|^3) + O(|\mathcal{T}_o|K^2) + O(|\mathcal{T}_o|K) = O|\mathcal{T}_o|^3K^2)$$

In order, these terms correspond to computing $P(f_k(\mathcal{T}_o), P(W_{k,k'}(\mathcal{T}_o)), W(t)f(t)$ and the y terms.

- **Computing the expectation of the q term:** To handle the q term we decompose it as the entropy of $(K + K^2)|\mathcal{T}_o|$ one-dimensional Gaussian terms. This is $O(1)$ per term so the complexity is $O(|\mathcal{T}_o|K^2)$
- **Sampling**: Sampling a one dimensional normal distribution is $O(1)$. Therefore, as we have $|\mathcal{T}_o|(K + K^2)$ variables to sample from the complexity is $O(|\mathcal{T}_o|K^2)$.
- **Total**: Adding all of these together we conclude that the total complexity of the forward computation is $O(K^2|\mathcal{T}_o|^3)$ per gradient step.

Complexity of Partially Factorized Family If we use the partially factorized family the number of parameters increases due to the covariance matrices. Each term now takes $O(|\mathcal{T}_o|^2)$ space rather than $O(|\mathcal{T}_o|)$ as before. This is a significant increase in memory that makes this family slower slower. However, the complexity is the same, although with worse constant factors. We detail the reasoning below.

- **Computing the log probability**: This is exactly the same as before.
- **Computing the expectation of the q term**: We now decompose $\mathbb{E}_q[\log q]$ into $K + K^2$ terms each corresponding to a gaussian process. Each of these terms takes usually $O(|\mathcal{T}_o|^3)$ to compute as the entropy is given by

$$-\frac{1}{2}\ln|\Sigma| + \frac{T}{2}(1 + \ln(2\pi))$$

where Σ is the covariance matrix. However, computing the determinant can be done in $O(|\mathcal{T}_o|)$ time with the cholesky decomposition. Therefore, computing the entropy takes $O(|\mathcal{T}_o|K^2)$

- **Sampling**: For the sampling step we have to sample $K + K^2$ gaussian processes. This is usually $O(|\mathcal{T}_o|^3)$ but we can use the cholesky decomposition to make it more efficient. In detail, we have to sample $(K^2 + K)|\mathcal{T}_o|$ standard normal distributions, and then use the cholesky decomposition alongside the trick that if X is standard normal $LX + \mu$ is normal with mean μ and covariance $LL^\top$. Putting this together we end up with a complexity of $O(|\mathcal{T}_o|^2 K^2)$, which is an extra factor of $\mathcal{T}_o$.

- **Total**: The total complexity of the forward computation is just as bad as before. However, memory is much worse and the sampling becomes much less efficient.

F.2 Computing

Computing the term $P(\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u | \mathcal{F}_o, \mathcal{W}_o)$ involves sampling $\mathcal{F}_o, \mathcal{W}_o$ from the distribution above, fitting one GP per coordinate and then drawing samples from each GP. We ignore the initial sampling step because we accounted for it above an focus on the last two here.

First, fitting one GP per coordinate is $O(|\mathcal{T}_o|^3)$ and there are $K^2 + K$ such coordinates, therefore the total complexity of the initial step is $O(|\mathcal{T}_o|^3 K^2)$. After fitting each GP, evaluating a new point has complexity $O(|\mathcal{T}_o|^2)$. Because we want to evaluate $|\mathcal{T}_u|$ points for each of the coordinates we get a bound for this step $O(K^2 T^2 |\mathcal{T}_u|)$. Putting all of this together, one draw of $\mathcal{Y}_u, \mathcal{W}_u, \mathcal{F}_u$ is

$$O(K(|\mathcal{T}_o|^3 + |\mathcal{T}_u||\mathcal{T}_o|^2))$$

In practice, we repeat the second step multiple times to avoid paying the large $|\mathcal{T}_o|^3$ term.