



Article

Generating hard-to-obtain information from easy-to-obtain information: applications in drug discovery and clinical inference

Matthew Amodio¹, Dennis Shung², Daniel Burkhardt³, Patrick Wong⁴, Michael Simonov⁴, Yu Yamamoto⁴, David van Dijk⁵, Francis Perry Wilson⁴, Akiko Iwasaki⁶⁷, and Smita Krishnaswamy^{31*}

¹ Yale University, Department of Computer Science

² Yale University School of Medicine, Department of Internal Medicine

³ Yale University School of Medicine, Department of Genetics

⁴ Yale University School of Medicine, Clinical and Translational Research Accelerator, Department of Medicine

⁵ Yale University School of Medicine, Department of Cardiology

⁶ Yale University School of Medicine, Department of Immunobiology

⁷ Howard Hughes Medical Institute

* Correspondence: smita.krishnaswamy@yale.edu

Version August 20, 2020 submitted to Cell Patterns

1 **Abstract:** In many important contexts involving measurements of biological entities, there are distinct
2 categories of information: some information is easy-to-obtain information (EI) and can be gathered
3 on virtually every subject of interest, while other information is hard-to-obtain information (HI) and
4 can only be gathered on some of the biological samples. For example, in the context of drug discovery,
5 measurements like the chemical structure of a drug are EI, while measurements of the transcriptome
6 of a cell population perturbed with the drug is HI. In the clinical context, basic health monitoring
7 is EI because it is already being captured as part of other processes, while cellular measurements
8 like flow cytometry or even ultimate patient outcome are HI. We propose building a model to make
9 probabilistic predictions of HI from EI on the samples that have both kinds of measurements, which
10 will allow us to generalize and predict the HI on a large set of samples from just the EI. To accomplish
11 this, we present a conditional Generative Adversarial Network (cGAN) framework we call the Feature
12 Mapping GAN (FMGAN). By using the EI as conditions to map to the HI, we demonstrate that
13 FMGAN can accurately predict the HI, with heterogeneity in cases of distributions of HI from EI. We
14 show that FMGAN is flexible in that it can learn rich and complex mappings from EI to HI, and can
15 take into account manifold structure in the EI space where available. We demonstrate this in a variety
16 of contexts including generating RNA sequencing results on cell lines subjected to drug perturbations
17 using drug chemical structure, and generating clinical outcomes from patient lab measurements.
18 Most notably, we are able to generate synthetic flow cytometry data from clinical variables on a cohort
19 of COVID-19 patients—effectively describing their immune response in great detail, and showcasing
20 the power of generating expensive FACS data from ubiquitously available patient monitoring data.

21 **Keywords:** generative adversarial networks; drug perturbations; conditional generative models

22 **Bigger Picture:** Many experiments face a trade-off between gathering easy-to-collect information
23 on many samples or hard-to-collect information on a smaller number of small due to costs in terms of
24 both money and time. We demonstrate that a mapping between the easy-to-collect and hard-to-collect
25 information can be trained as a conditional GAN from a subset of samples with both measured. With
26 our conditional GAN model known as Feature-Mapping GAN (FMGAN), the results of expensive
27 experiments can be predicted, saving on the costs of actually performing the experiment. This can have

28 major impact in many settings. We study two example settings. First, in the field of pharmaceutical
29 drug discovery early phase pharmaceutical experiments require casting a wide net to find a few
30 potential leads to follow. In the long term, development pipelines can be re-designed to specifically
31 utilize FMGAN in an optimal way to accelerate the process of drug discovery. FMGAN can also have
32 a major impact in clinical setting, where routinely measured variables like blood pressure or heart rate
33 can be used to predict important health outcomes and therefore deciding the best course of treatment.

34 **1. Introduction**

35 When collecting information on biological entities, for example hospital patients, cells, or drugs,
36 we are often faced with the choice of collecting easy-to-obtain information (EI) on many entities or
37 collecting hard-to-obtain information (HI) on a few entities. For example, in a drug library of millions
38 of drugs, it is easy to obtain chemical structure information but hard to obtain RNA sequencing
39 information of cells treated with drugs. On patients, it may be easy to obtain information such as heart
40 rate and lab values, but hard to obtain blood flow cytometry information. Here, we present a neural
41 network-based method that can bridge the gap between these sources of information on entities like
42 drugs or patients.

43 We introduce a framework based on a conditional Generative Adversarial Network (cGAN) that
44 we call Feature Mapping GAN (FMGAN), which learns a mapping from EI to a distribution of HI.
45 The FMGAN takes in noise as input, the EI information as the *condition* and the HI as the output. For
46 instance, given the chemical structure of a drug, we can build a mapping to the RNA sequencing of
47 cells under the drug. Here, the EI is the chemical structure and is used as the condition for the cGAN.
48 Corresponding HI is then produced by the generator of the cGAN. We showcase this in many settings
49 involving different information obtained on drugs and patients.

50 Our use of a GAN-based framework is motivated by our applications' having complex,
51 one-to-many relationships between the EI and the HI. To illustrate this further, consider a simple linear
52 mapping between an EI variable and an HI variable. The linearity guarantees that small changes in the
53 EI will result in a small change in the HI, i.e. the mapping is *smooth*. However, with chemical structure,
54 for example, this is known not to be true: a small change in chemical structure can lead to vastly
55 different properties of a drug. Non-linear mappings can also be simple, such as a simple threshold
56 decision: if a particular clinical variable completely determines patient outcome, a logical decision
57 with a threshold would suffice. However, clinical outcomes are the result of complex couplings
58 between large groups of variables. This necessitates a rich mechanism of mapping EI to HI, capable of
59 representing the necessary complexity. Moreover, the mapping has to be stochastic. Since, it is unlikely
60 that the EI has complete information about the drug or patient in question, it is important for each
61 EI condition to be able to map to a range or a distribution of HI conditions. For example, replicates
62 of a drug perturbation experiment result in different gene expression results even when applied on
63 the same cell line [1]. This stochastic response can only be captured by a generative model that can
64 produce stochastic output. As GANs learn complex mappings from a random noise space (and, for
65 cGANs, an EI space) to the HI space, they have the required complexity and stochasticity. And with
66 their flexible training paradigm, they do so without having to make strong assumptions like those
67 involved in choosing a parametric family for the form of the HI distribution.

68 One of our motivating examples through this paper is the **drug discovery process**. A major part
69 of pharmacological research is devoted to drug discovery, where a large number of drug compounds
70 have to be sorted to find a small number of promising candidates [2]. This search can be guided by
71 information about the drug itself, as well as by the past history of how other drugs have performed [3].
72 By looking for drugs similar to ones that have shown success previously, promising candidates with
73 improved toxicity or efficacy can be identified. Improvements in this form of research, called hit-to-lead,
74 can save significant time and money. The search for promising candidate drugs is a daunting task, since
75 the state space of molecular libraries is in the millions, and possible drugs is in the tens of thousands
76 or more [4].

77 Here we specifically consider measurements involving drug perturbations, a commonly used
78 technique for measuring the effect of a drug [5–7]. We utilize drug perturbation data from the L1000
79 Connectivity Map dataset [1]. Perturbation involves introducing the drug to a sample of cells and
80 then measuring the gene expression of those cells after the drug treatment. By comparing the gene
81 expression of the cells before and after drug treatment, researchers can infer information about what the
82 drug does and how it works. Because perturbing a cell line with a drug involves physically performing
83 an experiment, including obtaining the cells, applying the drug, and getting the sequencing results, this
84 process can be expensive and time-consuming. We use the FMGAN to generate the RNA-sequencing
85 results from the drug structure to speed this process up by not having to perform all of the experiments
86 exhaustively. If only a subset of the drugs have a priori RNA-sequencing measurements, the rest can
87 be generated with the FMGAN, obviating the need for additional experimentation on a large number
88 of candidates.

89 Another motivating setting is that of **clinical data**. In the clinical setting, some measurements
90 are readily available EI, either because they are already measured as part of the standard patient
91 monitoring, or because they are non-invasive and do not pose any risk. We work with two clinical
92 datasets of this type. The first is an Electronic Intensive Care Unit (eICU) Collaborative Research
93 Database dataset, which includes as EI standard clinical measurements such as body temperature,
94 heart monitoring, and standard blood work [8,9]. With this EI we generated predicted clinical mortality,
95 a measurement whose value can normally only be obtained too late to act upon. Rather than due
96 to financial expense, this measurement is hard-to-obtain because it is irreversible, involving patient
97 mortality. With the FMGAN, predictions can be accurately generated from the EI and thus preventative
98 measures can be taken while positive interventions are still possible.

99 The second clinical dataset we work with uses similar clinical measurements as EI, but this time
100 on COVID-19 patients from Yale New Haven Hospital. In this case, the HI information are future
101 single-cell flow cytometry measurements from samples gathered on some of the patients. In practice,
102 these types of single-cell measurements cannot be performed exhaustively on every patient in the
103 clinic, for reasons of cost as well as time sensitivity. Thus, we use the FMGAN to be able to generate
104 future flow cytometry data which depicts compartments of the immune system from readily available
105 clinical data. With the FMGAN, we are then able to generate flow cytometry data for any number of
106 patients who only have clinical measurements available. This can be valuable as immune responses
107 have been shown to be highly predictive of mortality in COVID-19 [10].

108 In each of these datasets we not only utilize the natural flexibility of the cGAN in mapping, but
109 also explicitly design mechanisms for the cGAN to take advantage of any structure that *does exist* in the
110 EI. While EI-HI mapping is rarely linear or simple, there are many instances in which the HI is smooth
111 and respects geometric or manifold structure in the EI—which can be explicitly represented. Here, we
112 show two ways of taking into account latent structure in the EI. The first is by embedding the EI into
113 lower dimensional manifold-intrinsic coordinates, such as with the PHATE dimensionality-reduction
114 method, which has been shown to preserve manifold affinity [11]. We show this on the case of drug
115 perturbations where we measure some genes on perturbed cell lines, and impute the other genes. Since
116 the underlying cellular manifold measured is the same, both measured and withheld genes should
117 respect this structure. We also show this on clinical data where ICU measurements are embedded
118 with PHATE and then the embeddings are used to impute clinical outcomes. The second situation,
119 rather than embedding EI with PHATE, is to use a convolutional neural network to find a latent space
120 embedding of the data. We use this encoding of the EI where it is the chemical structure of the drug.
121 Here, we create a rich set of convolutional features of the chemical structure by treating it as an image.
122 In particular a small change in the structure can be reflected as larger changes in convolutional filter
123 outputs, and thus the latent space has more regularity with respect to the mapping than the original
124 chemical structure space.

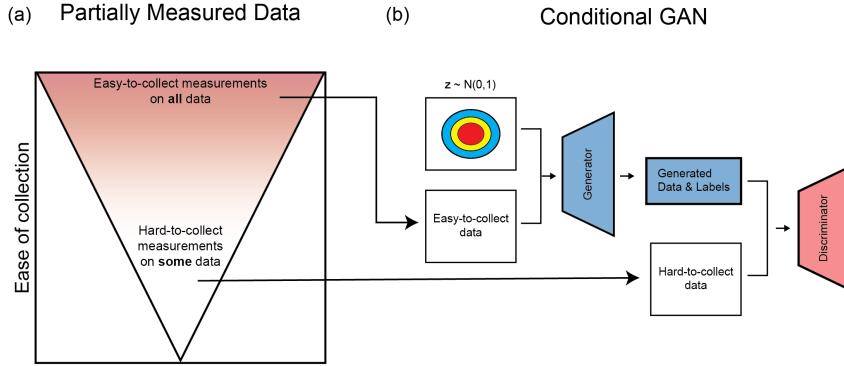


Figure 1. (a) The measurements on data are separated into “easy-to-collect information” (EI) and “hard-to-collect information” (HI). The easy-to-collect measurements are available on all data, while the hard-to-collect measurements are only available on some data. (b) With a Conditional GAN, we can learn to model the relationship between these two categories of measurements.

125 2. Results

126 2.1. FMGAN

127 The FMGAN we propose uses a conditional Generative Adversarial Network (cGAN) to generate
128 hard-to-collect information (like sequencing results from a perturbation experiment) from other
129 easy-to-collect information (like basic information on the drugs used). Specifically, we propose a cGAN
130 with the easy-to-collect information as conditions and the hard-to-collect information as the data
131 distribution. A cGAN is a generative model that learns to generate points based on a conditional label
132 that is given to the generator G . In the adversarial learning framework, G is guided into generating
133 realistic data during training by another network, the discriminator D , that tries to distinguish between
134 samples from the real data and samples from the generated data. The generator G and discriminator
135 D are trained by alternating optimization of G and D .

136 A standard GAN learns to map from random stochastic input $z \sim N(0, 1)$ (or a similarly simple
distribution) to the data distribution by training G and D in alternating gradient descent with the
following objective:

$$\min_G \max_D \mathbb{E}_{x \sim P(x)} \log(D(x)) + \mathbb{E}_{z \sim P_z} \log(1 - D(G(z)))$$

137 The generator in a cGAN receives both the random stochastic input z and a conditional label l and
thus has the following objective:

$$\min_G \max_D \mathbb{E}_{x_l \sim P(x|l)} \log(D(x|l)) + \mathbb{E}_{z \sim P_z} \log(1 - D(G(z|l)))$$

138 The cGAN was originally used in image generation contexts, where the condition referred to
what type of image should be generated (e.g. a dog). The cGAN is useful in this context because
the generator G receives a sample from a noise distribution (as in a typical GAN) as well as the
condition. Thus, it is able to generate a *distribution* that is conditioned on the label, as opposed to a
single deterministic output conditioned on the label. In the original use case, it can learn to generate a
wide variety of images of dogs when given the conditional label for dogs, for example. While many
previous methods exist for generating a single output from a single input, there are few alternatives for
generating a distribution of outputs from a single input without placing assumptions on the parametric
form of the output distribution.

139 The framework of the FMGAN is summarized in Figure 1. The columns of the data are separated
140 into easy-to-collect information (EI) and hard-to-collect information (HI). In the notation of the GAN,
141 we use the EI as the conditional label l and the HI as the data x . For observations that have both, we

148 train the FMGAN with the generator receiving a label l and a noise point z , while the discriminator
149 receives the label l and both real points x and the generated points $G(z|l)$. Then, after training, the
150 generator can generate points for conditions l without known data x . This allows us to impute HI
151 where we only have EI.

152 The FMGAN architecture is designed to take advantage of complex relationships between the
153 condition space and the data space. A single underlying entity (e.g. a drug or a patient) has a
154 representation in both spaces. In the EI space, the drug is a point, while in the HI space the drug is
155 represented by a distribution of cells perturbed by it. Despite the difference in structure, the FMGAN
156 is able to leverage regularities in the relationship between the two spaces. This relies on the FMGAN
157 being able to leverage manifold structure inherent within each space (for more discussion of manifold
158 structure, please see the supplementary information).

159 In some cases, the data modality for the EI is difficult to utilize: for example, the chemical
160 structure of the drug. The chemical structure can be represented as a string sequence called SMILES or
161 a two-dimensional image of the structure diagram. Small changes in the chemical structure can have
162 large changes on its function, but may appear to be minor changes to the overall SMILES string or
163 the overall structure diagram image. Thus, we use an embedding neural network, parameterized as
164 a convolutional network, to process these representations into a more regular space where standard
165 distances and directions are meaningful. This parameterization is crucial, as originally the structure is
166 not linear (or else simpler models could leverage it). But with convolutional networks, small changes in
167 the input can cascade down into deeper layers in complex ways and make potentially large, meaningful
168 shifts in the embedding. We further detail the architecture and design of this network in the Methods
169 section.

170 2.2. Modeling drug perturbation experiments

171 We first demonstrate the results of our FMGAN model on data from the L1000 Connectivity Map
172 (CMap) dataset [1]. The CMap dataset contains a matrix of genes by count values on various cell
173 lines under different drug perturbations. We examine the A375 cell line, a cell line from a human
174 diagnosed with malignant melanoma. In this densely measured dataset, we have all gene expression
175 measurements for each drug. Each drug also has various numbers of replicates of the same experiment.
176 These replicates produce variable effects, motivating the need for a framework that is capable of
177 modeling such stochasticity.

178 We design four separate experiments with this dataset:

- 179 1. A proof-of-concept that the cGAN framework can effectively model and predict gene expression
180 values when the conditions are known to be meaningful because they are selected holdout genes
181 from the expression matrix itself.
- 182 2. An experiment where the conditions are taken from a non-linear dimensionality reduction
183 method applied to the expressions.
- 184 3. A test of the full FMGAN pipeline where conditions represent chemical structure in the form of
185 SMILES strings, and thus embeddings for conditions must be learned.
- 186 4. A variation of the chemical structure conditions where they are represented as images of structure
187 diagram.

188 In each dataset, the measurement we choose for evaluation is maximum mean discrepancy
189 (MMD) [12]. We choose this because we require a metric that is a distance between distributions, not a
190 distance merely between points. Taking the mean of a distance between points would not capture the
191 accuracy of any moments in the desired distribution beyond the first one. For the experiments based on
192 drug metadata (the SMILES strings and the chemical structure images experiments), we consider the
193 drug's distribution to be all of the gene profiles from that drug. For the experiments with conditions
194 derived from each gene profile (the heldout genes and dimensionality-reduction experiments), we
195 take a neighborhood of drugs around each condition and compare the predicted distribution of gene
196 profiles for those drugs with the true distribution.

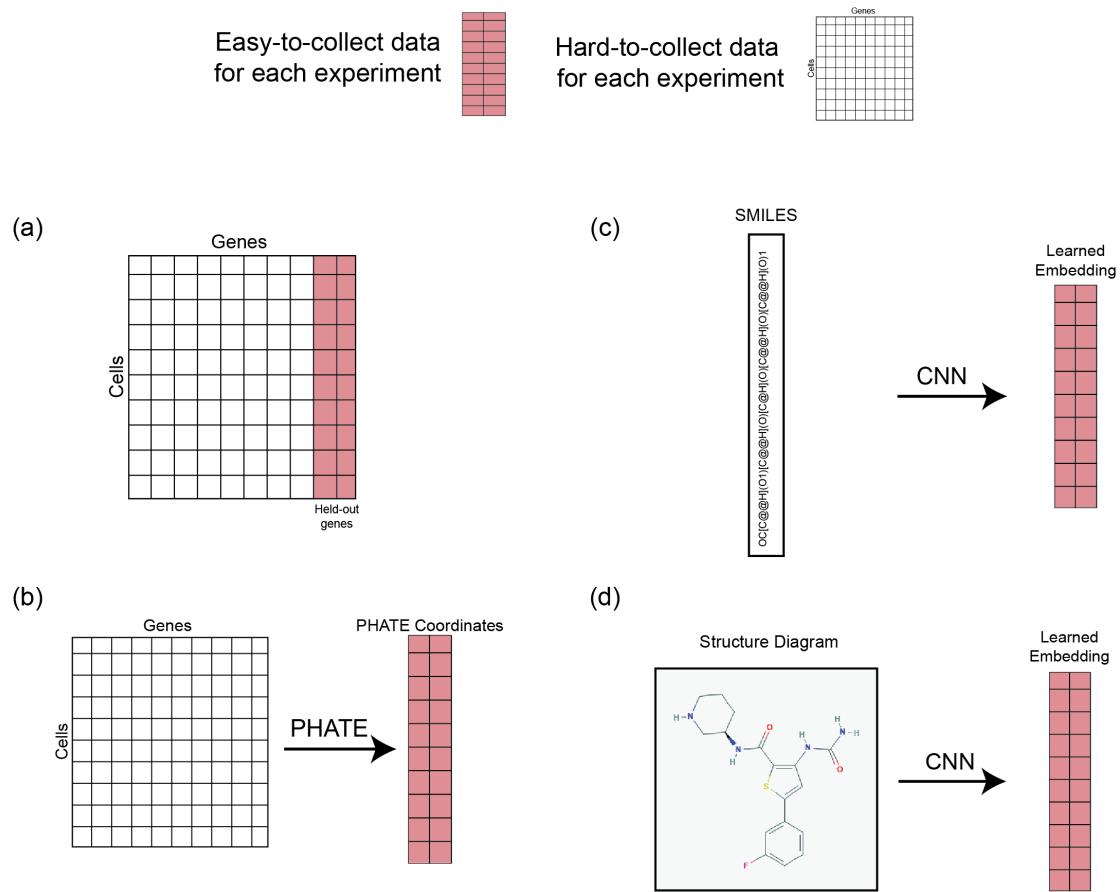


Figure 2. The formation of easy-to-collect (red columns) and hard-to-collect (white columns) data for each experiment with drug perturbation data. (a) in the held-out genes experiment, the easy-to-collect measurements are taken from held-out genes (b) in the PHATE coordinate experiment, they are the result of running on the genes matrix (c) in the SMILES string experiment, the easy-to-collect data is an embedding from processing this representation with a CNN (d) in the structure diagram experiment, it is the same as in the SMILES string experiment except run on the structure diagrams.

197 We compare our FMGAN to a baseline not built off of the cGAN framework. In developing a
198 baseline, we must compare to a model that takes in a point and outputs an entire distribution. As
199 most existing work yields deterministic output, we create our own stochastic distribution yielding
200 model to compare to. This model, which we term simply “Baseline”, takes a condition and a sample
201 from a random noise distribution as input, just like our FMGAN. However, unlike our model which
202 uses adversarial training and a deep neural network, the Baseline is a simpler, feed forward neural
203 network that minimizes the mean-squared-error (MSE) between the output of a linear transformation
204 and the real gene profile for that condition. As it is given noise input as well as a condition, it is still
205 able to generate whole distributions as predictions for each condition, rather than deterministic single
206 points. As generating conditional distributions (especially based off of oddly structured conditions like
207 images or strings) is relatively understudied in the computational biology field, we find no directly
208 comparably published methods that can be applied to this problem, thus necessitating our creating
209 Baseline.

210 2.2.1. Predicting gene expression under drug perturbation

211 To show our cGAN can learn informative mappings from the EI space to the gene expression
212 space, as distinct from the rest of the process, we first choose a means of obtaining EI that are known

213 to be meaningfully connected to the gene expression space. Specifically, we artificially hold out ten
214 genes and use their values as EI, with the GAN tasked with generating the values for all other genes.

215 This experimental design is summarized in Figure 2a. We choose the ten genes algorithmically by
216 selecting one randomly and then greedily adding to the set the one with the least shared correlation
217 with the others, to ensure the information in their values have as little redundancy as possible: PHGDH,
218 PRCP, CIAPIN1, GNAI1, PLSCR1, SOX4, MAP2K5, BAD, SPP1, and TIAM1. In addition to dividing
219 up the gene space to use these ten genes to predict all of the rest, we also divide up the cell space and
220 train on 80% of the cell data, with the last 20% heldout for testing.

221 We find our cGAN is able to successfully leverage information in the EI space to accurately model
222 the data. We designed our proof of concept deliberately so that the true values are known for each gene
223 expression and drug we ask our network to predict. These values can be compared to the predictions
224 with MMD for a measure of accuracy.

225 Our cGAN is able to generate predictions with an MMD of 2.847 between it and the validation
226 set (drugs it has never previously seen), showing it very effectively learned to model the dependency
227 structure between the EI space and the HI space, even on newly introduced drugs (Table 1). This is in
228 comparison to the Baseline model, which has a higher (worse) MMD of 2.922. It is noteworthy that the
229 FMGAN outperforms the baseline even in this case, where no processing of the EI needs to take place,
230 as they are numerically meaningful values to begin with.

231 We also can visualize the embedding spaces learned by the generator to investigate the model.
232 Shown in Figure 3a are the generator's embeddings colored by each of the heldout genes. As we can
233 see, the generator found some of these more informative in learning an EI embedding than others. We
234 can quantify this by building a regression model to try to predict the value of each gene given the
235 embedding to determine the most valuable of the heldout genes. By this measure, PHGDH, PRCP, and
236 GNAI1 are the most important genes. Analyzing the embeddings in this way is useful for determining
237 which part of the EI space was most informative for generating the HI space, and we will continue to
238 do this with more complex EI in later experiments.

239 2.2.2. PHATE coordinates as conditions for manifold-structured EI

240 Our next experiment formulates the EI space not as individual heldout genes, but instead on a
241 dimensionality-reduced representation of the whole space. We theorize that this approach would be
242 beneficial over the previous held-out-genes experiment if the EI data exhibits manifold structure. If
243 it does, this processing will have made a geometric representation of the EI that corresponds to the
244 HI, and thus the mapping is computationally simpler. Previous work has shown that gene expression
245 profiles often do exhibit this manifold structure [11,13,14].

246 We run the embedding tool PHATE on the gene profiles to calculate two coordinates, which we
247 then use as EI in our FMGAN [11]. Doing so preserves the manifold structure of the data, allowing
248 for a meaningful transformation to the HI space. This process is depicted in Figure 2b. As usual, we
249 separate cells into an 80%/20% training/testing split for evaluation purposes, after being subsampled
250 to ten thousand points for computational feasibility with the dimensionality reduction method, and
251 we report scores on the evaluation points.

252 As shown in Table 1, once again the FMGAN better models the target distribution, as measured
253 by MMD between its predictions in the neighborhood of each point and the true values. The FMGAN's
254 predictions had an MMD of 0.179, compared to the baseline MMD of 0.330 (a 45.7% improvement). It
255 is also interesting to note that although the MMDs are not directly comparable across the experiments
256 (because the target distribution is changing each time, from all drugs in neighborhood around a
257 coordinate to all drugs with the same metadata), the PHATE coordinates provide the most accurate
258 predictions.

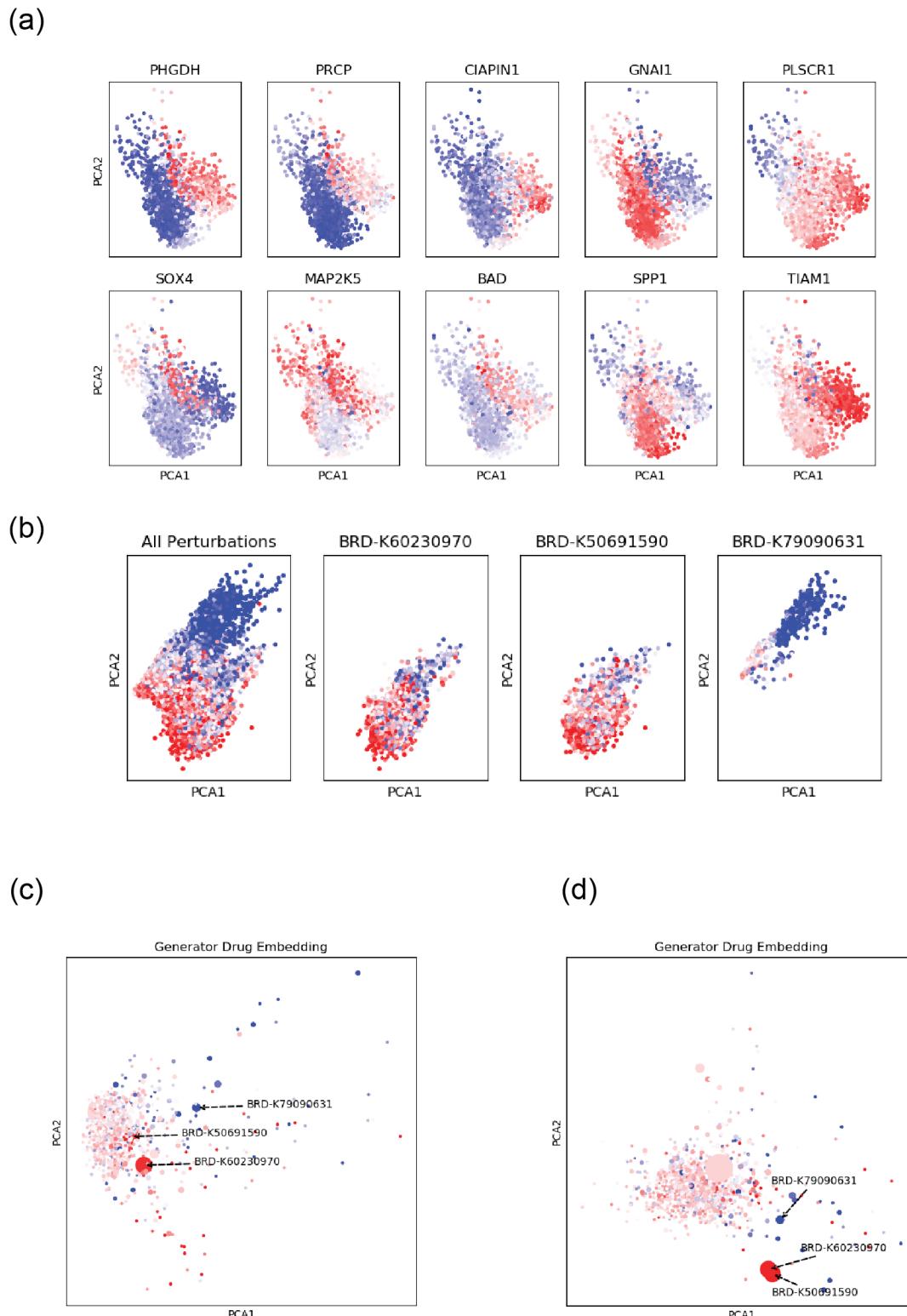


Figure 3. (a) Visualization of the embedding of *cells* in the held-out genes experiment, colored by each held-out gene. The network has inferred the structure of the space from these genes. (b) The raw data, colored by the expression of gene EIF4G2, separated into the three most abundant drugs: BRD-K60230970, BRD-K50691590, and BRD-K79090631. (c) The generator's embedding space of *drugs* from the SMILES strings experiment, with the same three drugs highlighted. The embedding in shows that the drugs with similar distributions have been embedded into similar locations in the learned embedding space. (d) The same as in (c) but with the structure diagram experiment.

259 2.2.3. Predicting Gene Expression from Drug Chemical Structure

260 Next, we test the full pipeline of FMGAN by using SMILES string embeddings as the EI
261 (summarized in Figure 2c). This is a much more challenging test case, because in the previous
262 cases each point in HI space had a distinct condition, and in the case of the PHATE coordinates, that
263 condition was derived from the data it had to predict. In this case, many different data points have the
264 same condition, and thus the relationship is much less direct between the EI and the HI.

265 An additional wrinkle also arises in this setting where the conditions to the cGAN are learned
266 from a raw data structure, rather than *a priori* existing in their final numerical form like heldout genes
267 or PHATE coordinates. Since G and D are trained adversarially and each depends on the embedder E ,
268 the networks could try to beat each other by manipulating the embeddings into being non-informative
269 for the other network. Thus, we let G and D learn their own embedder E , thus removing the incentive
270 to make E non-informative.

271 As in the previous experiment, we separate the data into an 80%/20% training/testing split for
272 evaluation purposes, but this time split along the drugs since each condition gives rise to many points
273 in the HI space. Table 1 indicates that the FMGAN had an MMD of 1.191 compared to the baseline of
274 1.510 (a 21.1% improvement).

275 EI Space Analysis

276 In this section, we investigate further the EI space learned from the SMILES strings by the
277 generator. In the two previous experiments, the conditions given to the FMGAN had information
278 more readily available, either in the form of raw data or even more informative PHATE coordinates.
279 The SMILES strings, by contrast, must be informatively processed for the learned conditions to be
280 meaningful.

281 In this learned EI space, there is one condition coordinate for each drug (while the HI consists of
282 many perturbations from each drug). Shown in Figure 3b is the raw data colored by the value of gene
283 EIF4G2. Then, all of the perturbations from each of three drugs are shown separately: BRD-K60230970,
284 BRD-K50691590, and BRD-K79090631. As we can see, the first two are characterized by high expression
285 of this gene and are quite similar to each other. The third, however, is quite distinct, in a separate space
286 of the embedding, and is characterized by much lower expression of this gene.

287 We compare this to the embedding learned by the generator, which we show in Figure 3c. In this
288 plot, each drug is one point, colored by the mean gene value of all perturbations for that drug and
289 with a point whose size is scaled by the number of perturbations for that drug. We see that the first
290 two drugs are in the central part of the space, and closer to each other than they are to BRD-K79090631.
291 The drug BRD-K79090631 is off in a different part of the space, along with other drugs low in EIF4G2.
292 This shows that the learned conditions from the generator have indeed identified information about
293 the drugs and taken complex sequential representations and mapped them into a much simpler space.

294 2.2.4. Predicting gene expression from drug structure diagrams

295 The final experiment we consider for the drug perturbation data is the formation of the condition
296 space from an image representation of the chemical structure (Figure 3d). These images are downloaded
297 from the PubChem PUG REST API [15]. An example image for the drug BRD-U86686840 is shown
298 in Figure 2d. They are given as input to a two-dimensional CNN designed for image processing, as
299 points in the original $h \times w \times c$ pixel space, with $h = w = 64$ and $c = 3$. While a CNN is used in both
300 the SMILES string case and this one, the underlying data is in a fundamentally different structure. As
301 in the SMILES string experiment, both the generator and the discriminator learned their own CNN to
302 develop embeddings adversarially.

303 Table 1 shows that the FMGAN performed slightly better with these chemical structure diagrams
304 as compared to the SMILES strings (1.177 MMD). The baseline model scored significantly worse
305 with these images as compared to the SMILES strings. This illustrates the FMGAN's flexibility,

MMD Scores	Heldout Genes	PHATE Coordinates	SMILES	Chemical Structure Image
FMGAN	2.847	0.179	1.191	1.177
Baseline	2.922	0.330	1.510	1.798

Table 1. MMD scores (lower is better) across all datasets for the drug data for both models. The FMGAN more accurately predicts the distribution from each condition for all methods of forming the condition space.

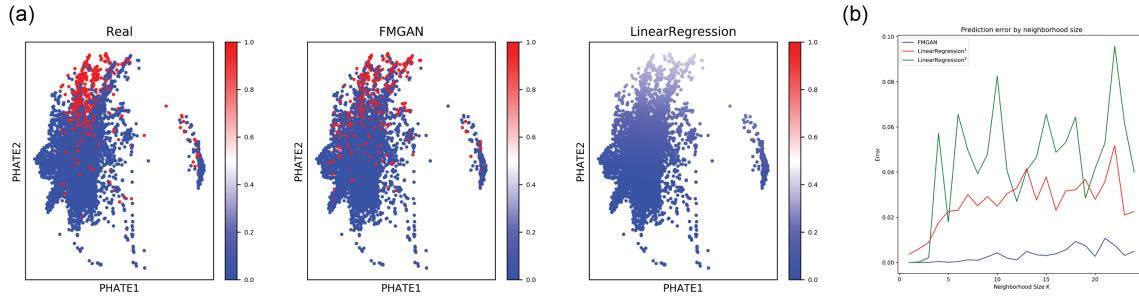


Figure 4. (a) The raw data from the eICU clinical outcomes experiment, along with FMGAN generated data and a linear regression baseline. (b) Quantitative evaluation of the model and the baseline.

306 as it performs comparably with drastically different structures (a long one-dimensional string as
 307 opposed to a natural image). That the chemical structure images perform slightly better is perhaps
 308 a sign that two-dimensional image convolutional networks are currently more effective at distilling
 309 this information than one-dimensional sequence convolutional networks, but the FMGAN's flexible
 310 framework allows it to keep improving with advances in deep learning architectures. Another
 311 possibility is that the structure diagrams have relevant information more easily separable from
 312 irrelevant information, making them an easier statistical task.

313 EI Space Analysis

314 In Figure 3d, we show the learned embedding from the generator. We color the embedding by
 315 the same gene and highlight the same three drugs as in the previous experiment: BRD-K60230970,
 316 BRD-K50691590, and BRD-K79090631. As before the learned conditions have taken a space where it
 317 is hard to characterize the information it contains (raw images in pixel space) and mapped them to
 318 a simpler space with numerically meaningful points. This can be seen by noting that the two drugs
 319 with similar distributions in the raw data (BRD-K60230970 and BRD-K50691590) have been mapped to
 320 nearly identical conditions, while they are separate from the drug with a very different distribution
 321 (BRD-K79090631). In fact, this goes towards an explanation of the improvement in performance over
 322 the SMILES string model, as the embedder has placed the drugs with similar distributions closer to
 323 each other in conditions, making the generator's job easier.

324 2.3. Predicting clinical outcomes

325 We demonstrate the versatility of our proposed method by experimenting on data in a very
 326 different context from the drug perturbations of the previous section. Here we work on clinical data
 327 from two different datasets. In each case, we use data derived from clinical measurements on patients
 328 to predict their clinical outcomes.

329 2.3.1. Predicting eICU clinical outcomes

330 For our first clinical experiment, we use data from patients at high risk for mortality due to
 331 severe illness, selected from the eICU Collaborative Research Database [8,9]. As conditions for the

332 FMGAN, we use measurements that are components of the in the widely-used APACHE score.
333 The APACHE score predicts mortality from age, immunocompromised status, heart measures, and
334 respiratory measures [16]. We pass these features through PHATE to develop conditions and then
335 predict mortality as our response variable. For more details on the data and pre-processing, please see
336 the Supplementary Information.

337 Figure 4 shows the real data, which is noisy but still shows different density of mortality in
338 different parts of the space. We also see the FMGAN generated data next to it: qualitatively, these
339 predictions resemble the raw data to a substantial degree. As a baseline, we can build a linear
340 regression model that tries to predict this response variable as a function of the coordinates. Due
341 to the probabilistic nature of the response, the linear regression predicts a low chance of mortality
342 everywhere in the space, with a slight uptick in probability in the dense region.

343 This is different from our FMGAN, which better models the binary nature of the output: in each
344 region there are some zeros and some ones as opposed to every point having a small constant value
345 like 0.1. To quantify the accuracy of each model, we have to develop an evaluation criterion that
346 looks at different regions and compares the true number of mortalities and predicted number in that
347 region. This metric assumes that within each local neighborhood, which point gets which label is
348 partially determined by randomness, and that the true signal is the proportion of points within that
349 neighborhood. Using this metric, we can compute the prediction error as the difference between the
350 predicted number of mortalities in a neighborhood and the true number.

351 Specifically, we compute K partitions of the data using the nearest neighbors clustering algorithm.
352 In each neighborhood, we compare the proportion of positive predictions (using a threshold of 0.5) and
353 the proportion of real positive outcomes with a mean-squared error measurement. Figure 4b shows
354 this for varying numbers of neighborhoods K. Also, while our model naturally outputs data like the
355 underlying data and thus has an easily identifiable threshold of 0.5 for a positive prediction, the linear
356 regression does not have an obvious choice for a threshold for a positive prediction. We use both the
357 default 0.5 (labeled LinearRegression¹) and the i^{th} percentile of the output, where i is chosen to match
358 the total proportion of real responses equal to one and the predicted proportion responses equal to
359 one.

360 Figure 4b shows a chart of these values for increasing numbers of neighborhoods to divide the
361 space into. The errors for the linear regression models range from 0.02 to 0.10 depending on the
362 neighborhood size, while the FMGAN remains below 0.01 for all neighborhood sizes. This means
363 the regression model has at least doubled the error of the FMGAN in each neighborhood size. The
364 stochasticity in the data makes it so that the GAN framework, which incorporates stochastic noise
365 input, is best able to generate output like the real data.

366 2.3.2. Predicting COVID-19 clinical outcomes

367 In this section we present an experiment that learns a mapping between clinical measurements
368 and FACS measurements from COVID-19 patients [17]. The clinical measurements are taken from the
369 first 24 hours in the ICU, with a patient's record being the most extreme value taken during that period
370 when more than one record is taken. To test the ability of FMGAN to make practical, and actionable
371 predictions we learn to generate the first flow cytometry measurement, taken from anywhere from the
372 first week to the eleventh week of the stay. Thus, we model *future* flow cytometry with *present* clinical
373 data.

374 The conditions we use for the FMGAN, as in the previous experiment, are PHATE coordinates of
375 embedded clinical variables. In the PHATE embedding each patient is represented by a a vector of
376 variables, listed in the supplement in Table 3. For each of 129 patients, we also have matched FACS
377 measurements on 14 proteins obtained from each patient, which are listed in the supplement in Table 4.
378 While the clinical measurements are relatively easy and inexpensive to obtain, FACS samples are
379 comparatively expensive and time-consuming to obtain. Thus, we wish to learn a model that can
380 accurately generate FACS data from a patient's clinical measurements alone.

Generating FACS Data from Clinical Measurements Conditions

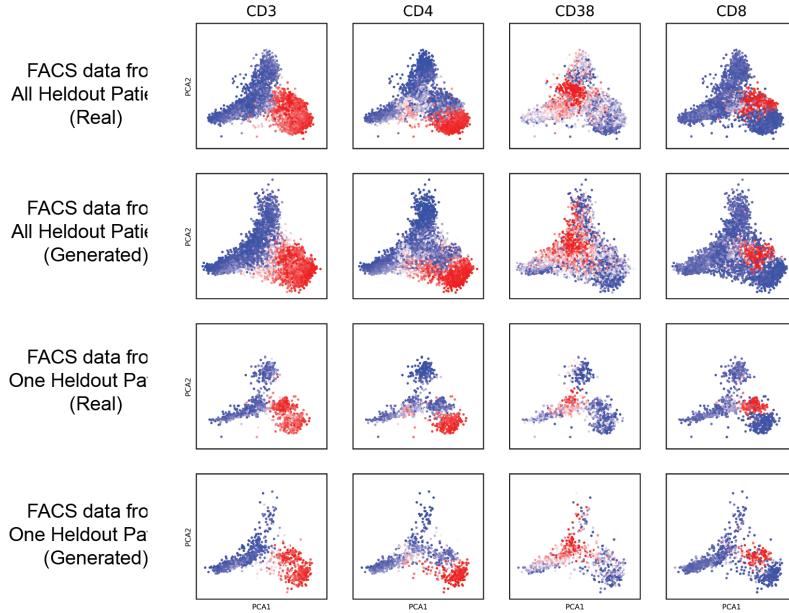


Figure 5. FACS data generated from clinical measurements in the COVID-19 data. Top row: for all 13 held-out patients, the real FACS measurements. Second row: for all 13 held-out patients, generated FACS measurements from the FMGAN. Third row: a single patient’s real FACS measurements. Bottom row: a single patient’s generated FACS measurements.

381 To evaluate the ability of the FMGAN to perform this generation, we train on 90% of the patients
382 (116) and withhold 10% of the patients (13) for evaluation. We train to generate a distribution of FACS
383 measurements from each single condition corresponding to a patient’s clinical measurements. In
384 Figure 5, we see the resulting data from all 13 heldout patients in the top row. In the second row, we
385 see the corresponding FMGAN generated data. Remarkably, the FMGAN learned to accurately model
386 the true distribution of FACS data even for the never-before-seen patients. Distinct populations of
387 cells are visible: CD3+ T cell populations including both CD4+ (T helper cells) and CD8+ (Cytotoxic T
388 cells), as well as a CD38+ population. With each protein marker, the FMGAN accurately models the
389 underlying data distribution.

390 In the bottom two rows of Figure 5, we see the FMGAN model the distribution from a single
391 patient accurately, as well. This per-patient generation forms the basis for our quantification of the
392 model’s accuracy. We utilize the same baseline as in the previous section. For each patient, we
393 measure the distribution distance between the predicted distribution and the true distribution of
394 FACS data (scored by MMD, as before). Table 2 shows the FMGAN is able to produce distributions
395 very close to the true underlying distribution for each patient, while the baseline model does not. As
396 each distribution is complex with many different cell populations with varying proportions, it is not
397 surprising that the more richly expressive FMGAN is better able to model the true data.

398 We note that with the FMGAN, we are able to predict the FACS measurements on
399 never-before-seen patients, based on their clinical measurement alone. However, this relied upon
400 the patients in the training set being representative of the patients in the held-out set. In practical
401 applications, this means that the population of patients would need to be chosen carefully and diversely
402 for the predictions to be meaningful for future patients.

MMD Scores	COVID-19 FACS data
FMGAN	0.022 +/- 0.008
Baseline	0.898 +/- 0.015

Table 2. MMD scores (lower is better) on the COVID-19 data, with mean and standard deviation across the 13 held-out patients. The FMGAN outperforms the baseline significantly.

403 **3. Methods**

404 *3.1. Conditional Generative Adversarial Networks*

405 In a Generative Adversarial Network (GAN), samples from the generator G can be obtained
406 by taking samples from $z \sim Z$ and then performing the forward pass with the learned weights of
407 the network. But while the values of z control which points G generates, we do not know how to
408 ask for specific types of points from G (more discussion of the original, unconditional GAN is in the
409 Supplementary Information).

410 The lack of this functionality motivated the need for the conditional GAN (cGAN) framework [18,
411 19]. The cGAN augments the standard GAN by introducing label information for each point. These
412 labels stratify the total population of points into different groups. The generator is provided a given
413 label in addition to the random noise as input, and the discriminator is provided with not only real
414 and generated points, but also the labels for each point. As a result, the generator not only learns to
415 generate realistic data, but it also learns to generate realistic data *for a given label*.

416 After training, the labels, whose meaning is known to us, can be provided to the generator to
417 generate points of a particular type on demand. Because G is provided both a label and a random
418 sample from Z , the cGAN is able to model not just a mapping from a label to a single point, but instead
419 a mapping from a label to an entire distribution.

420 Expressing the cGAN formula mathematically yields a similar equation as to the original GAN,
421 except with the modeled data distributions being marginal distributions conditioned on the label l of
422 each point:

$$\min_G \max_D \mathbb{E}_{x_l \sim P(x|l)} \log(D(x|l)) + \mathbb{E}_{z \sim P_z} \log(1 - D(G(z|l)))$$

423 Learning a generative model conditioned on the labels allows information sharing across labels,
424 another advantage of the cGAN framework. Since the generator G must share weights across labels,
425 the signal for any particular label l_i is blended with the signal from all other labels $l_j, j \neq i$, allowing
426 for learning without massive amounts of data for each label.

427 *3.2. Chemical Structure and SMILES Strings*

428 Conditional GANs are a powerful construction for guided generation, but require some known
429 label space to be used. While the label space must be relevant to the measured data space for an
430 informative model to be learned, the relationship need not be simple and can be noisy. When the data
431 space is gene expression after a drug perturbation, as in our application here, one relevant source of
432 labels is metadata about the structure of the drug used for the perturbation. We consider two ways of
433 representing this structure for our label space: a one-dimensional sequence of letters called a Simplified
434 Molecular-Input Line-Entry System (SMILES) string, and a two-dimensional image called a structure
435 diagram.

436 *SMILES strings*

437 A SMILES string encodes the chemical structure of a drug in a variable-length set of standard
438 letters and symbols. Each character in the string represents an element of the chemical's physical

436 formation, for example an atom, a bond, or a ring. For example, the common molecule glucose has the
437 following structure:

438 OC[C@H](O1)[C@H](O)[C@H](O)[C@H](O)[C@H](O)[C@H](O)

439 The letters indicate elements oxygen, carbon, and hydrogen, with @ denoting stereochemical
440 configuration, and brackets and parentheses representing bonds and branches, respectively. Clearly,
441 while providing rich information about the drug, this representation does not immediately lend itself
442 to use as a condition. In order to distill these variable-length sequences into a fixed-size representation
443 where similar structures have similar representations, we use a sequence-encoding neural network to
444 embed each structure into a latent space.

445 **Structure diagram**

446 An alternate way of representing chemical structure, more intelligible for a human observer than
447 SMILES strings, is a structure diagram. These have letters representing elements as in the SMILES
448 strings, but also are distinguished by colors, while different types of bonds are indicated with simple
449 lines. These images are downloaded from the PubChem PUG REST API [15]. While specifying how
450 to get information about the structure out of this image explicitly would be impossible (in terms of
451 RGB pixels), a neural network can learn how to process these images itself in order to accomplish
452 its training objective, all through a completely differentiable optimization with stochastic gradient
453 descent.

454 **3.3. FMGAN**

455 We describe the architecture for the FMGAN in this section. In the SMILES strings experiment, to
456 obtain a fixed-length D_E -dimensional vector for each string, we represent each input as a sequence
457 of length N_{seq} vectors, with N_{seq} being the longest SMILES string in the database. Each element
458 in the sequence is a vector representing the character in that position of the sequence (with a null
459 token padding the end of any sequence shorter than N_{seq}). As is standard in language processing, we
460 learn character-level embeddings simultaneously with the sequence-level processing. Let V be the
461 vocabulary, or set of all characters. The character-level embeddings are rows of a $|V| \times D_{char}$ matrix W ,
462 where $|V|$ is the number of characters in the vocabulary and D_{char} is a hyperparameter, the size of the
463 character embedding. Each input is then represented as a sequence where the i^{th} element is the row of
464 W corresponding to the i^{th} character in the SMILES string.

465 The size of the vocabulary (number of characters including start, end, and null tokens) is 43. We
466 chose the size of the character-level embedding to be 100. The embedder network E consists of two
467 convolutional layers with 64 and 32 filters, respectively, each with a kernel-size of 40 and stride-length 2
468 with batch normalization and a leaky ReLU activation applied to the output. These convolutional layers
469 are followed by four fully-connected layers which gradually reduce the dimensionality of the data with
470 400, 200, 100, and 50 filters, respectively. All layers except the last one have batch normalization and
471 leaky ReLU activations. The generator and discriminator have the same architecture as the previous
472 experiment.

473 This input representation is then passed through E , a convolutional neural network (CNN), which
474 produces the sequence embeddings. E performs one-dimensional convolutions over each sequence
475 followed by fully-connected layers, eventually outputting a single D_E -dimensional vector for each
476 SMILES string. We let these embeddings form the condition space for the next stage in FMGAN, the
477 conditional GAN.

478 For the structure diagram experiment, we start with images that are points in $h \times w \times c$ space, with
479 $h = w = 64$ and $c = 3$. They are then processed with a CNN. The CNN consists of four convolutional
480 layers with stride 2, kernel size 3, and filters of 32, 64, 128, and 256, respectively. Batch normalization
481 and a ReLU activation was used for each layer. Finally, after the convolutions, one fully connected

482 layer maps the flattened output to a 100-dimensional point, representing the embedding learned for
483 the particular diagram.

484 For both experiments, the generator structure, after the drugs are processed into conditions, is
485 the same. Let c_i be the condition for drug i formed by the embedder. Let x_i be the D_x -dimensional
486 corresponding gene expression profile from a perturbation experiment performed with drug i . We
487 build a GAN that trains a generator G to model the underlying data distribution conditioned upon the
488 structure $p_{data}(x|c)$. G takes as input both a sample from a noise distribution (we choose an isotropic
489 Gaussian) $z \sim Z$, and a condition c_i . G maps these inputs to a D_x -dimensional point. Then, the
490 discriminator D takes both a D_x -dimensional point and a condition c and outputs a single scalar
491 representing whether it thinks the point was generated by G or was a sample from p_{data} . These
492 networks then train in the standard alternating gradient descent paradigm of GANs previously
493 detailed.

494 For specific hyperparameter choices and data dimensionality details, we refer to the
495 Supplementary Information.

496 We note a few additional points about the FMGAN framework. First, since everything in the
497 network including the character-level embeddings, the embedder E , and the GAN are all expressed
498 differentiably, the whole pipeline can be trained at once in an end-to-end manner. Thus, the
499 character-level embeddings and the convolutional weights can be optimized for producing SMILES
500 strings embeddings *useful for this specific task and context*. This is a powerful consequence, as defining
501 what makes a good static embedding of a high-dimensional sequence may be ambiguous without
502 reference to a particular task.

503 4. Discussion

504 The FMGAN model allows us to predict hard-to-obtain information for samples where we only
505 directly measure easy-to-obtain information. We demonstrate that the FMGAN can accurately model
506 never-before-seen samples in these contexts. In the drug discovery context, this allows the potential
507 impact of saving on expense and time by not performing as many physical experiments and instead
508 modeling their results. In the clinical context, this allows for the modeling of patient data sooner, with
509 more time to take positive interventions.

510 Furthermore, the flexible framework of the cGAN we develop for the FMGAN allows for EI that
511 requires advanced processing to be used as the conditional input. We demonstrate this on images
512 and long one-dimensional sequences, but this can be extended to other difficult-to-represent data. For
513 example, in the clinical setting, the advances in natural language processing achieved by deep neural
514 networks could be utilized to process doctor's notes as raw text and then incorporated into the model.

515 We demonstrate that the FMGAN is able to leverage structure in the condition space in both
516 manifold form (from the PHATE coordinates) and discrete form (from chemical structure strings).
517 While seemingly similar, these are very different from an information theoretical point of view. In
518 the manifold setting, differences in input can create differences in output in a smooth way, but in the
519 discrete setting, one small change in an individual feature may have a large effect on the output while
520 another small change in a different feature has no effect on the output at all. For example, in a chemical
521 structure string, modifications to some locations will not change the function at all, while in other
522 locations a single change will determine function.

523 While we demonstrate that the FMGAN can be usefully applied to generative problems in a wide
524 variety of modalities, and, as we show, even in the presence of high amounts of stochasticity.

525

526 1. Subramanian, A.; Narayan, R.; Corsello, S.M.; Peck, D.D.; Natoli, T.E.; Lu, X.; Gould, J.; Davis, J.F.; Tubelli,
527 A.A.; Asiedu, J.K.; others. A next generation connectivity map: L1000 platform and the first 1,000,000
528 profiles. *Cell* **2017**, *171*, 1437–1452.

529 2. Vogel, H.G.; Vogel, W.H. *Drug discovery and evaluation: pharmacological assays*; Springer Science & Business
530 Media, 2013.

531 3. Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E.W. Computational methods in drug discovery. *Pharmacological reviews* **2014**, *66*, 334–395.

532 4. Hoffmann, T.; Gastreich, M. The next level in chemical space navigation: going far beyond enumerable
533 compound libraries. *Drug discovery today* **2019**, *24*, 1148–1156.

534 5. Haider, S.; Pal, R. Inference of tumor inhibition pathways from drug perturbation data. 2013 IEEE Global
535 Conference on Signal and Information Processing. IEEE, 2013, pp. 95–98.

536 6. Larsson, O.; Morita, M.; Topisirovic, I.; Alain, T.; Blouin, M.J.; Pollak, M.; Sonenberg, N. Distinct
537 perturbation of the translatome by the antidiabetic drug metformin. *Proceedings of the National Academy of
538 Sciences* **2012**, *109*, 8977–8982.

539 7. Korkut, A.; Wang, W.; Demir, E.; Aksoy, B.A.; Jing, X.; Molinelli, E.J.; Babur, Ö.; Bemis, D.L.; Sumer, S.O.;
540 Solit, D.B.; others. Perturbation biology nominates upstream–downstream drug combinations in RAF
541 inhibitor resistant melanoma cells. *Elife* **2015**, *4*, e04640.

542 8. Pollard, T.J.; Johnson, A.E.; Raffa, J.D.; Celi, L.A.; Mark, R.G.; Badawi, O. The eICU Collaborative Research
543 Database, a freely available multi-center database for critical care research. *Scientific data* **2018**, *5*, 180178.

544 9. Goldberger, A.L.; Amaral, L.A.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody,
545 G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research
546 resource for complex physiologic signals. *circulation* **2000**, *101*, e215–e220.

547 10. Lucas, C.; Wong, P.; Klein, J.; Castro, T.B.; Silva, J.; Sundaram, M.; Ellingson, M.K.; Mao, T.; Oh, J.E.;
548 Israelow, B.; others. Longitudinal analyses reveal immunological misfiring in severe COVID-19. *Nature*
549 **2020**.

550 11. Moon, K.R.; van Dijk, D.; Wang, Z.; Gigante, S.; Burkhardt, D.B.; Chen, W.S.; Yim, K.; van den Elzen, A.;
551 Hirn, M.J.; Coifman, R.R.; others. Visualizing structure and transitions in high-dimensional biological data.
552 *Nature Biotechnology* **2019**, *37*, 1482–1492.

553 12. Dziugaite, G.K.; Roy, D.M.; Ghahramani, Z. Training generative neural networks via maximum mean
554 discrepancy optimization. *arXiv preprint arXiv:1505.03906* **2015**.

555 13. Amodio, M.; Van Dijk, D.; Srinivasan, K.; Chen, W.S.; Mohsen, H.; Moon, K.R.; Campbell, A.; Zhao, Y.;
556 Wang, X.; Venkataswamy, M.; others. Exploring single-cell data with deep multitasking neural networks.
557 *Nature methods* **2019**, pp. 1–7.

558 14. Amodio, M.; Krishnaswamy, S. Magan: Aligning biological manifolds. *arXiv preprint arXiv:1803.00385*
559 **2018**.

560 15. PubChem. *PubChem PUG REST*, (accessed January 24, 2020).
561 <https://pubchemdocs.ncbi.nlm.nih.gov/pug-rest-tutorial>.

562 16. Knaus, W.A.; Draper, E.A.; Wagner, D.P.; Zimmerman, J.E. APACHE II: a severity of disease classification
563 system. *Critical care medicine* **1985**, *13*, 818–829.

564 17. Haimovich, A.; Ravindra, N.G.; Stoytchev, S.; Young, H.P.; Wilson, F.P.; van Dijk, D.; Schulz, W.L.; Taylor,
565 R.A. Development and validation of the COVID-19 severity index (CSI): a prognostic tool for early
566 respiratory decompensation. *medRxiv* **2020**.

567 18. Springenberg, J.T. Unsupervised and semi-supervised learning with categorical generative adversarial
568 networks. *arXiv preprint arXiv:1511.06390* **2015**.

569 19. Lotter, W.; Kreiman, G.; Cox, D. Unsupervised learning of visual structure using predictive generative
570 networks. *arXiv preprint arXiv:1511.06380* **2015**.

571 20. Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis.
572 *arXiv preprint arXiv:1809.11096* **2018**.

573 21. Yang, Z.; Chen, W.; Wang, F.; Xu, B. Unsupervised neural machine translation with weight sharing. *arXiv
574 preprint arXiv:1804.09057* **2018**.

575 22. Amodio, M.; Krishnaswamy, S. Travelgan: Image-to-image translation by transformation vector learning.
576 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8983–8992.

577 23. Gupta, A.; Zou, J. Feedback GAN (FBGAN) for DNA: a novel feedback-loop architecture for optimizing
578 protein functions. *arXiv preprint arXiv:1804.01694* **2018**.

579 24. Kodali, N.; Abernethy, J.; Hays, J.; Kira, Z. On convergence and stability of gans. *arXiv preprint
580 arXiv:1705.07215* **2017**.

582 25. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale
583 update rule converge to a local nash equilibrium. *Advances in neural information processing systems*,
584 2017, pp. 6626–6637.

585 26. Mescheder, L.; Geiger, A.; Nowozin, S. Which training methods for GANs do actually converge? *arXiv
586 preprint arXiv:1801.04406* 2018.

587 27. Houle, M.E. Dimensionality, discriminability, density and distance distributions. 2013 IEEE 13th
588 International Conference on Data Mining Workshops. IEEE, 2013, pp. 468–473.

589 28. Moon, K.R.; Stanley, J.S.; Burkhardt, D.; van Dijk, D.; Wolf, G.; Krishnaswamy, S. Manifold learning-based
590 methods for analyzing single-cell RNA-sequencing data. *Current Opinion in Systems Biology* 2018, 7, 36–46.
591 doi:10.1016/j.coisb.2017.12.008.

592 29. van Dijk, D.; Sharma, R.; Nainys, J.; Yim, K.; Kathail, P.; Carr, A.J.; Burdziak, C.; Moon, K.R.; Chaffer, C.L.;
593 Pattabiraman, D.; Bierie, B.; Mazutis, L.; Wolf, G.; Krishnaswamy, S.; Pe'er, D. Recovering Gene Interactions
594 from Single-Cell Data Using Data Diffusion. *Cell* 2018, 174, 716 – 729.e27. doi:10.1016/j.cell.2018.05.061.

595 30. Levine, J.; Simonds, E.; Bendall, S.; Davis, K.; Amir, E.a.; Tadmor, M.; Litvin, O.; Fienberg, H.; Jager, A.;
596 Zunder, E.; Finck, R.; Gedman, A.; Radtke, I.; Downing, J.; Pe'er, D.; Nolan, G. Data-Driven Phenotypic
597 Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell* 2015, 162, 184–197.
598 doi:10.1016/j.cell.2015.05.047.

599 31. Moon, K.R.; van Dijk, D.; Wang, Z.; Gigante, S.; Burkhardt, D.B.; Chen, W.S.; Yim, K.; Elzen, A.v.d.; Hirn,
600 M.J.; Coifman, R.R.; Ivanova, N.B.; Wolf, G.; Krishnaswamy, S. Visualizing structure and transitions in
601 high-dimensional biological data. *Nature Biotechnology* 2019, 37, 1482–1492.

602 32. Haghverdi, L.; Buettner, M.; Wolf, F.A.; Buettner, F.; Theis, F.J. Diffusion pseudotime robustly reconstructs
603 lineage branching. *Nature methods* 2016, 13, 845.

604 33. Coifman, R.R.; Lafon, S. Diffusion maps. *Applied and computational harmonic analysis* 2006, 21, 5–30.

605 34. van der Maaten, L.; Hinton, G. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine
606 Learning Research* 2008, 9, 2579–2605.

607 35. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension
608 reduction. *arXiv preprint arXiv:1802.03426* 2018.

609 36. Li, C.L.; Chang, W.C.; Cheng, Y.; Yang, Y.; Póczos, B. Mmd gan: Towards deeper understanding of moment
610 matching network. *Advances in Neural Information Processing Systems*, 2017, pp. 2203–2213.

611 37. Borgwardt, K.M.; Gretton, A.; Rasch, M.J.; Kriegel, H.P.; Schölkopf, B.; Smola, A.J. Integrating structured
612 biological data by kernel maximum mean discrepancy. *Bioinformatics* 2006, 22, e49–e57.

613 38. Cosgriff, C.V.; Celi, L.A.; Ko, S.; Sundaresan, T.; de la Hoz, M.Á.A.; Kaufman, A.R.; Stone, D.J.; Badawi, O.;
614 Deliberato, R.O. Developing well-calibrated illness severity scores for decision support in the critically ill.
615 *NPJ digital medicine* 2019, 2, 1–8.

616 **5. Software Availability**

617 <https://github.com/KrishnaswamyLab/FMGAN>

618 **6. Supplementary Information**

619 *6.1. Generative Adversarial Networks*

620 Generative Adversarial Networks (GANs) are a deep learning framework for learning a generative
621 model of a data distribution. In recent years, they have gained significant popularity by achieving
622 state-of-the-art performance on applications to images, language, sequences, and many other data
623 modalities [14,20–23]. GANs differ from other types of models by not using explicit likelihood
624 measures nor relying on having a meaningful distance measure between points. Instead, they teach a
625 generator neural network G with a second discriminator network D using the following equation:

$$\min_G \max_D \mathbb{E}_{x \sim P_x} [\log(D(x))] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]$$

626 where x is the training data, z is a noise distribution that provides stochasticity to the generator
627 and is chosen to be easy to sample from (typically an isotropic Gaussian).

628 *6.2. Conditional Generative Adversarial Networks*

629 Conditional Generative Adversarial Networks (cGANs) originated from the desire for having
630 greater control over generation from GANs. In the case where external information, such as class
631 labels, are available, we would like to be able to generate a random point from a specific class. The
632 methods devised to achieve this involve providing a random label to the generator during training
633 and then providing this label and the generated image to the discriminator. The discriminator also
634 receives real images and their labels, allowing it to learn their joint distribution.

635 Once the model has been trained in this way, control over generation can be used to generate a
636 point from a particular class by feeding the desired class into the generator. This process especially
637 benefits from having fine-grained, continuous conditions like we have, as this gives even more precise
638 control over generation.

639 *6.3. Optimization*

640 The networks G and D take turns optimizing their objectives through alternating gradient descent.
641 Throughout training, the discriminator provides gradient information to the generator guiding it to
642 better quality generation. This powerful framework provides the ability to model arbitrarily complex
643 distributions without making any explicit parametric or limiting assumptions about their shape.

644 Theoretical analysis of GANs have shown their ability to converge to an optimal point where the
645 generated distribution is indistinguishable from the true distribution [24–26]. The ability to converge
646 to this optimal generative model without specifying a distribution distance is especially helpful in our
647 applications, where the points lie in high dimensions and the curse of dimensionality makes distances
648 problematic [27].

649 **Manifold learning**

650 A useful assumption in representation learning is that high biomedical dimensional data originates
651 from an intrinsic low dimensional manifold that is mapped via nonlinear functions to observable high
652 dimensional measurements; this is commonly referred to as the manifold assumption. In particular,
653 we believe that since biological entities like patients, cells lie in lower dimensional spaces because
654 of informational redundancy and coordination between measured features (coordinating genes, or
655 coordinated combinations of residues on molecules). Further, we believe that these low dimensional

656 spaces form smoothly varying patches because of natural heterogeneity between entities. The fact that
657 the manifold model is successful in modeling biological entities has been shown in literature numerous
658 times [28] and has lead to successful methods data denoising [29], clustering [30], visualization [31],
659 and progression analysis [32].

660 Formally, let \mathcal{M}^d be a hidden d dimensional manifold that is only observable via a collection
661 of $n \gg d$ nonlinear functions $f_1, \dots, f_n : \mathcal{M}^d \rightarrow \mathbb{R}$ that enable its immersion in a high dimensional
662 ambient space as $F(\mathcal{M}^d) = \{\mathbf{f}(z) = (f_1(z), \dots, f_n(z))^T : z \in \mathcal{M}^d\} \subseteq \mathbb{R}^n$ from which data is collected.
663 Conversely, given a dataset $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^n$ of high dimensional observations, manifold
664 learning methods assume data points originate from a sampling $Z = \{z_i\}_{i=1}^N \in \mathcal{M}^d$ of the underlying
665 manifold via $x_i = \mathbf{f}(z_i)$, $i = 1, \dots, n$, and aim to learn a low dimensional intrinsic representation that
666 approximates the manifold geometry of \mathcal{M}^d .

667 To learn a manifold geometry from collected data, we use the popular diffusion maps construction
668 of [33] that uses diffusion coordinates to provide a natural global coordinate system derived from
669 eigenfunctions of the heat kernel, or equivalently the Laplace-Beltrami operator, over manifold
670 geometries. This construction starts by considering local similarities defined via a kernel $\mathcal{K}(x, y)$,
671 $x, y \in F(\mathcal{M}^d)$, that captures local neighborhoods in the data. We note that a popular choice for \mathcal{K} is the
672 Gaussian kernel $\exp(-\|x - y\|^2/\sigma)$, where $\sigma > 0$ is interpreted as a user-configurable neighborhood
673 size. However, such neighborhoods encode sampling density information together with local geometric
674 information. To construct a diffusion geometry that is robust to sampling density variations we use an
675 anisotropic kernel

$$\mathcal{K}(x, y) = \frac{\mathcal{G}(x, y)}{\|\mathcal{G}(x, \cdot)\|_1^\alpha \|\mathcal{G}(y, \cdot)\|_1^\alpha}, \quad \mathcal{G}(x, y) = e^{-\frac{\|x-y\|^2}{\sigma}}$$

676 as proposed in [33], where $0 \leq \alpha \leq 1$ controls the separation of geometry from density, with $\alpha = 0$
677 yielding the classic Gaussian kernel, and $\alpha = 1$ completely removing density and providing a geometric
678 equivalent to uniform sampling of the underlying manifold. Next, the similarities encoded by \mathcal{K} are
679 normalized to define transition probabilities $p(x, y) = \frac{\mathcal{K}(x, y)}{\|\mathcal{K}(x, \cdot)\|_1}$ that are organized in an $N \times N$ row
680 stochastic matrix

$$\mathbf{P}_{ij} = p(x_i, x_j) \tag{1}$$

681 that describes a Markovian diffusion process over the intrinsic geometry of the data. Finally, a
682 diffusion map [33] is defined by taking the eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ and (corresponding)
683 eigenvectors $\{\phi_j\}_{j=1}^N$ of \mathbf{P} , and mapping each data point $x_i \in X$ to an N dimensional vector $\Phi_t(x_i) =$
684 $[\lambda_1^t \phi_1(x_i), \dots, \lambda_N^t \phi_N(x_i)]^T$, where t represents a diffusion-time (i.e., number of transitions considered
685 in the diffusion process). In general, as t increases, most of the eigenvalues λ_j^t , $j = 1, \dots, N$, become
686 negligible, and thus truncated diffusion map coordinates can be used for dimensionality reduction
687 [33].

688 PHATE for structure-preserving visualization of Data

689 Several dimensionality reduction methods that render data into 2-D visuals like PCA and tSNE.
690 [34] and UMAP [35] exist. However, they often cannot handle the degree of noise in biomedical data.
691 More importantly, most of these methods are not constructed to preserve the global manifold structure
692 of the data. PCA cannot denoise in non-linear dimensions, tSNE/UMAP effectively only constrains
693 for near neighbor preservation—losing global structure. This motivated us to develop a method of
694 dimensionality reduction that retains both local and global structure, and denoises data [31].

695 PHATE also builds upon the diffusion-based manifold learning framework described above,
696 and involves the creation of a diffused Markov transition matrix from cellular data, as in MAGIC,
697 \mathbf{P}^t (Equation 1). PHATE collects all of the information in the diffusion operator into two dimensions
698 such that global and local distances are retained. To achieve this, PHATE considers the i th row

699 of \mathbf{P} as the representation of the i th datapoint in terms of its t -step diffusion probabilities to *all*
700 other datapoints. PHATE then preserves a novel distance between two datapoints, based on this
701 representation called *potential distance* ($pdist$). Potential distance is an M -divergence between the
702 distribution in row i , $\mathbf{P}_{i,:}^t$, and the distribution in row j , $\mathbf{P}_{j,:}^t$. These are indeed distributions as \mathbf{P}^t is
703 Markovian: $pdist(i, j) = \sqrt{\sum_k (\log(P^t(i, k) - P^t(j, k))^2}$.

704 The log scaling inherent in potential distance effectively acts as a damping factor which makes
705 faraway points similarly equal to nearby points in terms of diffusion probability. This gives PHATE
706 the ability to maintain global context. These potential distances are embedded with metric MDS as a
707 final step to derive a data visualization. We have shown that PHATE outperforms tSNE [34], UMAP
708 [35], force directed layout and 12 other methods on preservation of manifold affinity, and adjusted
709 rand index on clustered datasets, in a total of 1200 comparisons on synthetic and real datasets. In [31]
710 we also showcased the ability of PHATE to reason about differentiation systems and differentiation
711 trajectories in human embryonic cell development.

712 6.4. Maximum Mean Discrepancy

To evaluate the accuracy of the predicted distribution with respect to the true distribution for a given condition, we utilize maximum mean discrepancy (MMD) [36]. The MMD is a distribution distance based on a kernel applied to pairwise distances of each distribution. Specifically, MMD is calculated as:

$$MMD(X, Y) = \frac{1}{n} \sum_{i \neq i'} k(x_i, x_{i'}) + \frac{1}{m} \sum_{i \neq i'} k(y_i, y_{i'}) - \frac{2}{mn} \sum_{i \neq j} k(x_i, y_j)$$

713 for finite samples from distributions $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_n\}$ and kernel function k .
714 Two distributions have zero MMD if and only if they are equal. MMD has been used successfully
715 in biological systems in the past, particularly in detecting whether two systems were different in
716 distribution [37].

717 6.5. eICU Clinical data

718 A patient cohort at high risk for mortality due to severe illness was selected from the eICU
719 Collaborative Research Database, a public multicenter critical care database containing 200,859 ICU
720 admissions with 139,367 unique patients admitted to critical care units between 2014 and 2015 [8,9].
721 After excluding patients who did not have a calculated risk score for mortality, the APACHE IVa score
722 and who had at least one vital sign, the final dataset contained 146,587 encounters with 118,638 unique
723 patients. The structured datafields from automated vital signs, laboratory results, and treatments for
724 the patient cohort were extracted and transformed as described by a recent manuscript by taking
725 the most abnormal values in the first 24 hours from ICU admission and multiple imputation using
726 Bayesian Ridge Regression was used to fill missing variables [38].

Alanine Aminotransferase	Antibiotic	Aspartate Aminotransferase
Bilevel Positive Airway Pressure	Blood Urea Nitrogen	Chloride
Continuous Positive Airway Pressure	Creatinine	Ferritin
Glucose	High-Flow Nasal Cannula	High-sensitivity C-Reactive Protein
Hydroxychloroquine	Mechanical Ventilation	Nasal Cannula
Non-rebreather Mask	Oxygen Saturation	Procalcitonin
Respiratory Rate	Steroid	Systolic Blood Pressure
Tocilizumab	White Blood Cell Count	

Table 3. Clinical variables for the COVID-19 dataset.

CCR7	CD3	CD4	CD8	CD25	CD38	CD45RA
CD127	CXCR5	FSC	HLA-DR	PD1	SSC	TIM3

Table 4. Flow cytometry markers for the COVID-19 dataset.

727 6.6. COVID-19 Clinical data

728 The cohort of patients included only those who were hospitalized at any of 6 hospitals in the
729 Yale-New Haven Health System (Bridgeport, Greenwich, St. Raphael's Campus, Westerley, Lawrence
730 and Memorial, York Street Campus) during the period between March 1st, 2020 and June 1st, 2020
731 with a positive COVID test (nasopharyngeal source) between admission and discharge. Only the first
732 encounter was included in the dataset for patients with multiple encounters during the time period of
733 observation. Patients with a positive test prior to hospital admission but not tested during admission
734 or tested negative during admission were not included in the cohort. Data for these patients was
735 then extracted from the electronic health record (Epic, Verona, WI) and included data domains of
736 demographics (e.g. age and sex), medical history (e.g. history of diabetes), laboratory samples (e.g.
737 white blood cell count), as well as vital signs (e.g. blood pressure measurement). Pre-defined outcomes
738 included in-hospital mortality, transfer to the intensive care unit (ICU), as well as requirement for
739 invasive ventilation. In-hospital mortality was measured as patients being discharged from the hospital
740 with a deceased status. ICU care was measured through location data for patients and was manually
741 validated through chart review. Ventilation status was measured through procedure orders placed
742 during the patient's hospitalization and were validated through chart review.

743 Time-varying data, specifically vital signs as well as laboratory studies, were extracted at all
744 timepoints of measurement during a patient's admission.

745 Features were selected from a predictive model developed to predict early hospital respiratory
746 decompensation among patients with Covid-19 and augmented with treatment received. There were a
747 total of 19 clinical, laboratory, and treatment variables extracted: systolic blood pressure, respiratory
748 rate, oxygen saturation, blood urea nitrogen, creatinine, chloride, glucose, white blood cell count,
749 alanine aminotransferase, aspartate aminotransferase, high-sensitivity C-reactive protein, ferritin,
750 procalcitonin, age, gender, and treatment with hydroxychloroquine, steroid, antibiotic, or tocilizumab.
751 Only complete cases, or patients with recorded values for all 19 variables in the first 24 hours, were
752 included in the final dataset.

753 As preprocessing, the most abnormal value in the first 24 hours was selected for the clinical
754 and laboratory variables according to the methodology described in a previous electronic health
755 record-based study. The categorical variables for treatment were coded as binary (1 for received, 0 for
756 not recorded).

757 © 2020 by the authors. Submitted to *Cell Patterns* for possible open access publication
758 under the terms and conditions of the Creative Commons Attribution (CC BY) license
759 (<http://creativecommons.org/licenses/by/4.0/>).