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

Highlights
- Modeling RNA sequencing of cells perturbed with a drug with its chemical structure
- Modeling patients’ FACS from clinical monitoring data
- Enabling cGANs to learn their own condition space

Authors
Matthew Amodio, Dennis Shung, Daniel B. Burkhardt, ..., Francis Perry Wilson, Akiko Iwasaki, Smita Krishnaswamy

Correspondence
smita.krishnaswamy@yale.edu

In brief
Many experiments face a trade-off between gathering easy-to-collect information on many samples or hard-to-collect information on a smaller number of samples due to costs in terms of both money and time. With the model introduced here called the FMGAN, the results of expensive experiments can be predicted, saving on the costs of actually performing the experiment.
Generating hard-to-obtain information from easy-to-obtain information: Applications in drug discovery and clinical inference

Matthew Amodio, 1 Dennis Shung, 2 Daniel B. Burkhardt, 3 Patrick Wong, 4 Michael Simonov, 4 Yu Yamamoto, 4 David van Dijk, 5 Francis Perry Wilson, 4 Akiko Iwasaki, 6,7 and Smita Krishnaswamy 1,3,8,*

1Department of Computer Science, Yale University, New Haven, CT, USA
2Department of Internal Medicine, Yale University School of Medicine, New Haven, CT, USA
3Department of Genetics, Yale University School of Medicine, New Haven, CT, USA
4Clinical and Translational Research Accelerator, Department of Medicine, Yale University School of Medicine, New Haven, CT, USA
5Department of Cardiology, Yale University School of Medicine, New Haven, CT, USA
6Department of Immunobiology, Yale University School of Medicine, New Haven, CT, USA
7Howard Hughes Medical Institute
8Lead contact
*Correspondence: smita.krishnaswamy@yale.edu
https://doi.org/10.1016/j.patter.2021.100288

SUMMARY

Often when biological entities are measured in multiple ways, there are distinct categories of information: some information is easy-to-obtain information (EI) and can be gathered on virtually every subject of interest, while other information is hard-to-obtain information (HI) and can only be gathered on some. We propose building a model to make probabilistic predictions of HI using EI. Our feature mapping GAN (FMGAN), based on the conditional GAN framework, uses an embedding network to process conditions as part of the conditional GAN training to create manifold structure when it is not readily present in the conditions. We experiment on generating RNA sequencing of cell lines perturbed with a drug conditioned on the drug’s chemical structure and generating FACS data from clinical monitoring variables on a cohort of COVID-19 patients, effectively describing their immune response in great detail.

INTRODUCTION

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

We introduce a framework based on a conditional Generative Adversarial Network (cGAN) that we call Feature Mapping GAN (FMGAN), which learns a mapping from EI to a distribution of HI. The FMGAN takes in noise as input, the EI information as the condition, and the distribution of HI for that EI as the output. Unique to the FMGAN is an auxiliary network called the condition-embedding network. This network processes the EI in order to discover its latent manifold dimensions, from which the mapping can be smoother or more regular to learn. To illustrate the utility of this, consider a simple linear mapping between an EI variable and an HI variable. The linearity guarantees that small changes in the EI will result in a small change in the HI, i.e. a mapping is smooth. However, such a mapping is only possible if the EI is linearly related to the HI. With chemical structure, for example, this is known not to be true: a small change in chemical structure can lead to vastly different properties of a drug. Therefore, our condition-embedding network embeds or finds alternative coordinates for our conditions that lie in low-dimensional spaces (like lines or planes), thereby discovering latent structure that can render the mapping smooth. In the case of chemical structure, we use a convolutional neural network that performs complex mapping into manifold dimensions. This is key, as the addition of convolutional architectures allows the FMGAN to integrate conditions that have image structure with tabular, non-image transcriptomic measurements.

After the EI condition is appropriately processed to discover latent dimensions, it is passed to the generator to allow for a stochastic mapping. Since the HI is a distribution of cells in an RNA sequencing or fluorescence-activated cell sorting (FACS) sample, a model needs to generate data at the resolution of single cells given a condition, but also generate a variety of different cells for that individual condition. Moreover, in other cases it is unlikely that the EI has complete information about the drug or patient in question, and thus it is important for each EI condition to be able to map to a range or a distribution of HI conditions. For example, replicates of a drug perturbation experiment result in different gene expression results even when applied on the same cell line.1 This stochastic response can only be captured by a generative model that can produce stochastic output.

We showcase FMGAN on two main application areas, drug discovery and clinical prediction. Here we specifically consider measurements involving drug perturbations, a commonly used technique for measuring the effect of a drug.2-4 We use drug perturbation data from the L1000 Connectivity Map dataset.1 Because perturbing a cell line with a drug involves physically performing an experiment, including obtaining the cells, applying the drug, and getting the sequencing results, this process can be expensive and time-consuming. We use the FMGAN to generate the RNA sequencing results from the drug structure to speed this process up by not having to perform all of the experiments exhaustively. If only a subset of the drugs have a priori RNA sequencing measurements, the rest can be generated with the FMGAN, obviating the need for additional experimentation on a large number of candidates.

We also use these experiments to compare different ways of representing the chemical structure of drugs which we use as EI. We compare string and image-based representations, and by measuring how well the FMGAN models the HI data in each case, we can draw conclusions about the different representations. For example, we show that pictorial diagrams of chemical structure are more effective than string sequences when all other things are held equal.

Another motivating setting is that of clinical data. In the clinical setting, some measurements are readily available EI, either because they are already measured as part of the standard patient monitoring, or because they are noninvasive and do not pose any risk. The clinical dataset we work with uses as EI these clinical measurements from patients with coronavirus disease 2019 (COVID-19) from Yale New Haven Hospital. In this case, the HI includes future single-cell flow cytometry measurements from samples gathered on some of the patients. In practice, these types of single-cell measurements cannot be performed exhaustively on every patient in the clinic, for reasons of cost as well as time sensitivity. Thus, we use the FMGAN to be able to generate future flow cytometry data that depicts compartments of the immune system from readily available clinical data. With the FMGAN, we are then able to generate flow cytometry data for any number of patients who only have clinical measurements available. This can be valuable, as immune responses have been shown to be predictive of mortality in COVID-19.5

Related work

Conditional GANs

The FMGAN builds off of the cGAN framework, which differs from a regular GAN by learning to model data distributions that are conditioned upon a label that is supplied along with noise as input to the generator. In our FMGAN, we use the terminology “easy-to-obtain information” for the conditional label and “hard-to-obtain information” for the data distribution; we do this to emphasize the broad applicability of the FMGAN. In our applications, the EI is not a “label” for the HI in any classical sense: for example, it would be odd to call a patient’s present monitoring data a label for that patient’s future FACS data. Thus, our use of EI and HI is intended to help move beyond the scope of traditional cGAN applications.

Those traditional cGAN applications are usually similar to its first application: image generation. In image generation contexts, the condition referred to what type of image should be generated (e.g., a dog). The cGAN is able to generate a distribution that is conditioned on the label, for example, generating a wide variety of images of dogs when given the conditional label for dogs. Canonical use cases for cGANs use simple integer conditions of nonoverlapping classes, like the CIFAR dataset in which there are 10 distinct classes with 6,000 images per class, all known a priori.6 In cases like this, a human has already separated all images of one group from those of the others and the labels have no relation to each other: class 1 is not more similar to class 2 than it is to class 9 or 10. More importantly, this setting offers no ability to extrapolate. When trained on 10 conditional labels, no network is able to extrapolate to an unseen 11th condition because it does not have any information about how it relates to the conditions it trained on. This is because previous cGAN models do not model the condition space, instead only modeling each individual condition label separately. In the FMGAN, we model the condition space itself via the condition-embedding network. This allows for extrapolation to never-
before-seen conditions, using the functional model of the condition space it learned from the training data.

**Biological applications of cGANs**

Previous work has used cGANs for biological applications in ways that are related to, but in crucial ways that differ from our method. Some approaches have modeled single-cell transcriptomic data, as we do, but have predicted the expression of some target genes using the value of other landmark genes, not using any metadata separate from the expression matrix at all. Several other works used cell-type conditional labels, but have used integer conditions. These integers represent manually labeled cell types in some cases, or clusters identified from the transcriptomic data itself via an off-the-shelf clustering model. To the best of our knowledge, no previous work has learned conditional generation based on processing sequential or image representations of chemicals.

**Learning embeddings of biological data manifolds**

In this work, we build off of the observation that biological data in a high-dimensional ambient space often exhibits manifold structure in a low-dimensional latent space. We assume that the conditions in our framework form such a manifold, and that learning to generate off of their manifold coordinates leads to an easier mapping problem than doing so off of their ambient coordinates. In the FMGAN, we assume that the conditions form such a space. Much previous work has shown biological datasets can be meaningful represented as low-dimensional manifolds.

**RESULTS**

**FMGAN**

The FMGAN consists of a cGAN where the condition input is given by an auxiliary network called the condition-embedding network. This network that processes conditions is trained adversarially during the GAN training to best optimize the conditions for mapping between EI and HI.

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:

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

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

\[
\min_G \max_D \mathbb{E}_{x \sim P(x)} \log(D(x)) + \mathbb{E}_{z \sim P(z)} \log(1 - D(G(z, l)))
\]

**Condition-embedding network**

Condition-embedding networks on each side of the generator/discriminator adversarial process, a feature of the FMGAN that distinguishes it from previous models, allow for conditions to be used in a cGAN even when the information in the conditions is not in an easy-to-access form. This is because most biological data are known to have manifold structure.

Often, this manifold structure is not present in the original space in which the data exist, but instead exist in a latent lower-dimensional space. In the original data space for the EI, transformations from the EI to the HI may be highly nonsmooth and irregular. For example, consider shifting a particular element in a chemical formula. Shifting this element’s location by several spaces may result in no change or minor change to the molecule’s effect on a cell system. Then, a critical point is reached, when one more shift fundamentally alters the structure as observable by the perturbed cell system.

The condition-embedding network transforms the representations of these chemical structures into latent manifold representations (as seen in Figure 1). Small, smooth movements in these manifold coordinates results in small, smooth observable effects in the perturbed cell system. Thus, while in ambient data coordinates all shifts to the chemical structure are equal, in the manifold coordinates, only the ones that change the molecule’s function result in movement on the manifold.

Specifically, the condition-embedding network learns a mapping that produces a \( d_2 \)-dimensional embedding \( m \) from a \( d_1 \)-dimensional condition \( l \): \( \{l_1, l_2, \ldots, l_g\} \rightarrow (m_1, m_2, \ldots, m_d) \), where \( M_l = (m_1, m_2, \ldots, m_d) \) for condition \( l \). The function learned is such that \( M_1 - M_2 \) implies an empirical bound on the difference in generated distributions \( P_{M_1} \) and \( P_{M_2} \). We demonstrate this experimentally on our drug structure data in a later section.

The condition-embedding network, parameterized as either a fully connected neural network or as a convolutional neural network when the ambient data have structure that convolutions can process, uncovers the latent manifold and transforms points into these coordinates, subsequently passing them to the generator.

This rendering of the EI into manifold structure allows for generalization, i.e., it learns a landscape of the EI, for instance a landscape of drugs of which some examples of the drug effects...
on cells can be used to infer the effects of neighboring drugs (for more discussion of manifold structure, please see the supplemental information). After training the condition-embedding network, the FMGAN can generate from never-before-seen conditions, as it has a functional model of the condition space as well as the generated data space.

The framework of the FMGAN is summarized in Figure 1. The information on each data point is separated into EI and HI. In the notation of the GAN, we use the EI as the conditional label \( y \) and the HI as the data \( x \). For data points that have both, we process the condition and train the FMGAN with the generator receiving a label \( y \) and a noise point \( z \), while the discriminator receives the label \( y \) and both real points \( x \) and the generated points \( G(z|y) \). Then, after training, the generator can generate points for conditions \( i \) without known data \( x \). This allows us to impute EI where we only have HI.

### Modeling drug perturbation experiments

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

We design four separate experiments with this dataset:

1. A proof of concept that the cGAN framework can effectively model and predict gene expression values when the conditions are known to be meaningful because they are selected holdout genes from the expression matrix itself.
2. An experiment in which the conditions are taken from a nonlinear dimensionality-reduction method applied to the expressions, and thus do not need significant processing to make them a usable data manifold.
3. A test of the full FMGAN pipeline where conditions represent chemical structure in the form of Simplified Molecular-Input Line-Entry System (SMILES) strings, and thus do not provide information about the drug in a readily available numeric form, and meaningful embeddings for conditions must be learned.
4. A variation of the chemical structure conditions where they are represented as images of the chemical structure diagram as opposed to SMILES strings.

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

We make several comparisons to our FMGAN with each dataset. We first compare to a simpler model that takes in the condition and stochastic noise and minimizes mean-squared-error (MSE) between the output of a linear transformation and the real gene profile for that condition. As it is given noise input as well as a condition, it is still able to generate whole distributions as predictions for each condition, rather than deterministic single points. Secondly, we compare to a variational autoencoder (VAE) model that also receives the condition and must produce the real gene profile for that condition. The VAE then stochastically generates from its latent layer when needing to generate an entire distribution of points from a single condition. Finally, we compare two models just like the FMGAN but without one aspect of the full model (an ablation test). We compare to an FMGAN that does not use our novel condition-embedding network, and then we compare to an FMGAN that has the condition-embedding network but uses an MSE loss rather than an adversarial GAN loss. These ablation tests showcase the crucial role both the condition-embedding network and the adversarial training play in the overall FMGAN framework. We note in developing baselines, since generating conditional distributions (especially based off of oddly structured conditions like images or strings) is relatively understudied in the computational biology field, we find no directly comparably published methods that can be applied to this problem.

#### Predicting gene expression under drug perturbation

To show our FMGAN can learn informative mappings from the EI space to the gene expression space, as distinct from the rest of the process, we first choose a means of obtaining EI that is known to be meaningfully connected to the gene expression space. Specifically, we artificially hold out 10 genes and use their values as EI, with the FMGAN tasked with generating the values for all other genes.

This experimental design is summarized in Figure 2A. We choose the 10 genes algorithmically by selecting one randomly and then greedily adding to the set the one with the least shared correlation with the others, to ensure the information in their values have as little redundancy as possible: PHGDH, PRCP, CIAPIN1, GNAI1, PLSCR1, SOX4, MAP2K5, BAD, SPP1, and TIAM1. In addition to dividing up the gene space to use these 10 genes to predict all of the rest, we also divide up the cell space and train on 80% of the cell data, with the last 20% held out for testing.

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

As shown in Table 1, our FMGAN is able to generate predictions with the lowest MMD between them and the never-before-seen evaluation set (drugs it has never previously seen), showing it very effectively learned to model the dependency structure between the EI space and the HI space. The table reports the average and standard deviation of the MMD scores...
across five independent iterations. The FMGAN’s performance is significantly better in comparison with the other models, which have higher (worse) MMDs. It is noteworthy that the FMGAN outperforms the baselines even in this case, where we do not know a priori of any significant processing of the EI that needs to take place, as they are numerically meaningful values to begin with.

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

**PHATE coordinates as conditions for manifold-structured EI**

Our next experiment uses an EI space that consists of a dimensionality-reduced manifold representation of the data. While the whole FMGAN will take raw ambient data and learn a manifold simultaneously with learning to generate from the manifold, here we first experiment with generating from an already-learned manifold. We do this to test whether the FMGAN has the ability to accurately generate from a condition manifold as distinct from its ability to learn the manifold of the conditions, as well.

We theorize that this approach would be beneficial over the previous experiment of held-out genes if the gene space exhibits manifold structure, which previous work has shown is often the case.\(^1\)\(^2\),\(^1\)\(^6\),\(^1\)\(^7\) If so, and if the FMGAN is able to leverage the manifold structure, this processing will have made a geometric representation of the EI that corresponds to the HI, and thus the mapping is computationally simpler.

We run the embedding tool PHATE on the gene profiles to calculate two coordinates, which we then use as EI in our FMGAN.\(^1\)\(^7\) Doing so preserves the manifold structure of the data, allowing for a meaningful transformation to the HI space. This process is depicted in Figure 2B. As usual, we separate cells into an 80%/20% training/testing split for evaluation purposes, after being subsampled to 10,000 points for computational feasibility with the dimensionality-reduction method, and we report scores on the evaluation points.

As shown in Table 1, once again the FMGAN better models the target distribution, as measured by MMD between its predictions and the target distribution.
in the neighborhood of each point and the true values. The FMGAN’s predictions once again have a lower MMD than any of the alternative models. This is notable, as PHATE has already processed the conditions in the sequencing domain to produce lower-dimensional, more compact representations. The fact that the full FMGAN still performs the best indicates that optimal conditions for generating may be different from optimal conditions for visualizing or some other task, implying it is still beneficial to use the FMGAN’s condition-embedding network.

We observe the PHATE coordinates are much more effective than held-out genes as conditions. This is in line with our hypothesis that manifold structure that is related to the data space is beneficial for generating. Despite containing loosely correlated information, the held-out genes are too few and noisy, while the PHATE coordinates actually calculate a full data manifold from more genes and with more shared information with the ambient data that need to be modeled.

**Predicting gene expression from drug chemical structure represented as SMILES**

Next, we test the full pipeline of FMGAN by using SMILES strings as the EI (summarized in Figure 2C). This is a much more challenging test case than the previous ones, because in the previous cases each point in HI space had a distinct condition, and in the case of the PHATE coordinates, that condition was derived from the data it had to predict.

Most importantly, the conditions are in a raw data structure (one-hot vectors representing the SMILES strings token-by-token), rather than a priori existing in their final numerical form like held-out genes or scalar PHATE coordinates. This structure is not trivial to extract information from, as simple changes like one insertion shift the dimensions of every subsequent token, or the necessity of identifying recurring patterns that occur in different locations. These motivate the need for a convolutional condition-embedding network that looks for these kinds of structural forms and processes them prior to being given to the standard fully connected network that generates the RNA-sequencing data.

As in the previous experiment, we separate the data into an 80%/20% training/testing split for evaluation purposes, but this time split along the drugs since each condition gives rise to many points in the HI space. Table 1 indicates that the FMGAN has the lowest MMD of any of the models in this application. Perhaps unsurprisingly, there is also a larger gap between the full FMGAN and the FMGAN without a condition-embedding network than there was in the previous experiments. This provides verification that there is information about the RNA-sequencing that can be leveraged in the drug metadata, but special architectures are necessary to process them and access it.

**EI space analysis.** In addition to superior generative performance, we show the usefulness of having a generative model that learns embeddings by analyzing the learned EI SMILES strings embedding space. In this learned EI space, there is one condition coordinate for each drug (while the HI consists of many perturbations from each drug). Shown in Figure 3B are the raw data colored by the value of gene EIF4G2. Then, all of the perturbations from each of three drugs are shown separately: BRD-K60230970, BRD-K50691590, and BRD-K79090631. As we can see, the first two are characterized by high expression of this gene and are quite similar to each other. The third, however, is quite distinct, in a separate space of the embedding, and is characterized by much lower expression of this gene.

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

**Condition-to-generated data correlation.** The importance of the condition-embedding network can be seen by analyzing the distances between conditions and the distances between their corresponding generated data distributions. Since the generator must map from a condition to its generated distribution, the function it learns needs to be more complex (and will

| MMD scores | Held-out Genes | PHATE Coordinates | SMILES | Chemical Structure Image |
|-----------|----------------|------------------|--------|--------------------------|
| FMGAN     | 2.841 ± 0.006  | 0.213 ± 0.008    | 1.232 ± 0.005 | 1.219 ± 0.007 |
| (full model) |                |                  |        |                          |
| FMGAN     | 2.883 ± 0.005  | 0.220 ± 0.007    | 1.307 ± 0.006 | 1.621 ± 0.010 |
| (no condition-embedding) |                |                  |        |                          |
| FMGAN     | 2.956 ± 0.003  | 0.424 ± 0.005    | 1.482 ± 0.009 | 1.511 ± 0.011 |
| (no GAN)  |                |                  |        |                          |
| Linear    | 2.912 ± 0.006  | 0.533 ± 0.001    | 1.565 ± 0.005 | 1.772 ± 0.010 |
| VAE       | 2.962 ± 0.003  | 0.497 ± 0.002    | 1.886 ± 0.0035 | 2.012 ± 0.003 |

MMD scores (lower is better) across all datasets for the drug data for all models with mean and standard deviation reported across five independent runs. The full FMGAN with all of its components most accurately predicts the distribution from each condition for all methods of forming the condition space, although the datasets that require more advanced convolutional processing benefit the most.
Figure 3. Analysis of the generated data in ambient space, the embedded condition space, and the relationship between the two
(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.
generalize poorly) if nearby conditions induce very different distributions and faraway conditions induce very similar distributions. The condition-embedding network is able to take conditions from their original space, which may not be conveniently structured into a manifold space that is.

To test this, we perform the following experiment. For each pair of held-out drugs \(i\) and \(j\), we calculate the distance between condition \(l_i\) and \(l_j\), the distance between embedded conditions \(E(l_i)\) and \(E(l_j)\), and the MMD between the generated data from each condition \(G[E(l_i)]\) and \(G[E(l_j)]\).

As Table 2 shows, the correlation of distances between conditions and their MMDs between the generated data distributions is just 0.071. This is unsurprising, as the SMILES strings have structure that violates notions of Euclidean distances with respect to the underlying similarity of the drugs. Drugs that only differ by one element can be far very far from each other by Euclidean distance if the rest of the formula is shifted, but may produce almost identical data distributions if that inserted element has little effect on the overall structure. In the space that the condition-embedding network produces, the conditions form a manifold that is smooth with respect to the generated distributions. The correlation soared to 0.709, with distances in the embedded space meaningfully relating to how different their corresponding distributions are. These correlations can also be seen visually in the plotted data, as shown in Figure 3E. For further details and figures about this experiment, please refer to the supplement.

### Predicting gene expression from drug structure diagrams

The final experiment we consider for the drug perturbation data is the formation of the condition space from an image representation of the chemical structure (Figure 3D). These images are downloaded from the PubChem PUG REST API. An example image for the drug BRD-UB6686840 is shown in Figure 2D. They are given as input to a two-dimensional convolutional neural network (CNN) designed for image processing, as 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 the SMILES string case and this one, the underlying data are in a fundamentally different structure.

Table 1 shows that the FMGAN also outperformed the baseline models in this case, as before. However, something more deeply revealing is also apparent from the scores in this experiment as compared with the previous experiments.

The FMGAN scored slightly better with these chemical structure diagrams as compared with the SMILES strings. The baseline models, on the other hand, all scored significantly worse with the drug structure diagram images. For processing these images, clearly the convolutional condition-embedding network is necessary to achieve good generation performance. The full FMGAN is able to leverage the information in image form just as well as in a long one-dimensional sequence form, while other models (including the FMGAN without the condition-embedding network) are not.

This illustrates the FMGAN’s flexibility, as it performs comparably with such drastically different structures. That the chemical structure images perform slightly better may be a sign that two-dimensional image convolutional networks are currently more effective at distilling this information than one-dimensional sequence convolutional networks, but the FMGAN’s flexible framework allows it to keep improving with advances in deep-learning architectures. Another possibility is that the structure diagrams have relevant information more easily separable from irrelevant information, making them an easier statistical task.

### El space analysis.

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

### Condition-to-generated data correlation.

As in the SMILES string experiment, we evaluate the correlation of the distance between conditions and the distance between their data distributions. And just as in the previous case, the condition space is originally structured in such a way that distances are not meaningfully related to the effect that condition produces (images can be far away simply because they have the same diagram but it is rotated or shifted). The condition-embedding network produces a manifold that increases the correlation of these distances from 0.220 to 0.800 (Table 2).

We note that we can evaluate the two different representations of the chemical structure and compare them by looking at the scores in these experiments. The embedding with chemical structure images increases the correlation with MMD between ambient data distributions by 12% over the embedding with SMILES representations. In this latter case, the embedding network was slightly better able to organize the embedding space to align with the ambient data distributions. We caution careful interpretation of this, though: this improvement could stem from the images being a better representation of information, image-based embedding architectures being more

### Table 2. Correlation of condition space and generated ambient data space

| Correlation with Generated Data | Raw Conditions | Embedded Conditions |
|-------------------------------|----------------|---------------------|
| SMILES                        | 0.071 ± 0.002  | 0.709 ± 0.016       |
| Chemical Structure Images     | 0.220 ± 0.008  | 0.800 ± 0.012       |

(C) The generator’s embedding space of drugs from the SMILES strings experiment, with the same three drugs highlighted. The embedding 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.

(E) The conditions are more correlated to the generated data after they have been embedded.
Predicting flow cytometry data on COVID-19 patients

We demonstrate the versatility of our proposed method by experimenting on data in a very different context from the drug perturbations of the previous section. Here we work on clinical data that are derived from measurements taken early in the clinical stay and predict measurements that are taken later in the stay. Specifically, in this section, we present an experiment that learns a mapping between clinical measurements and FACS measurements from COVID-19 patients. The clinical measurements are taken from the first 24 h in the intensive care unit, with a patient’s record being the most extreme value taken during that period when more than one record is taken. To test the ability of FMGAN to generate practical, and actionable predictions, we learn to generate the first flow cytometry measurement, taken from anywhere from the first week to the 11th week of the stay. Thus, we model future flow cytometry with present clinical data.

The conditions we use for the FMGAN are PHATE coordinates of the present available clinical variables. In the PHATE embedding, each patient is represented by a vector of variables, listed in the supplement. For each of 129 patients, we also have matched FACS measurements on 14 proteins obtained from each patient, which are also listed in the supplement. While the clinical measurements are relatively easy and inexpensive to obtain, FACS samples are comparatively expensive and time-consuming to obtain. Thus, we wish to learn a model that can accurately generate FACS data from a patient’s clinical measurements alone. We note that while these conditions are in tabular form and processed with PHATE, we still use the full FMGAN to learn embeddings, as this achieved the best generative performance even for conditions of this form.

To evaluate the ability of the FMGAN to perform this generation, and to handle the relatively few number of distinct patients, we perform K-fold cross-evaluation, training each time on 80% of the patients (103) and withholding 20% of the patients (26) for evaluation. We train to generate a distribution of FACS measurements from each single condition corresponding to a patient’s clinical measurements. In Figure 4, we see the resulting data from all held-out patients in the top row from the first fold. In the second row, we see the corresponding FMGAN-generated data. Remarkably, the FMGAN learned to accurately model the true distribution of FACS data even for the never-before-seen patients. Distinct populations of cells are visible: CD3+ T-cell populations including both CD4+ (T-helper cells) and CD8+ (cytotoxic T cells), as well as a CD38+ population. With each protein marker, the FMGAN accurately models the underlying data distribution.

In the bottom two rows of Figure 4, we see the FMGAN model the distribution from a single patient accurately, as well. This per-patient generation forms the basis for our quantification of the model’s accuracy. We use the same baselines as in the previous section. For each fold (reported separately), and for each patient within that fold, we measure the distribution distance between the predicted distribution and the true distribution of FACS data (scored by MMD, as before). These numbers are reported in Table 3. That evaluation shows the FMGAN is able to produce distributions very close to the true underlying distribution for each patient, while the baseline models do not. As each distribution is complex with many different cell populations with varying proportions, it is not surprising that the more richly expressive FMGAN is best able to model the true data.

We note that with the FMGAN, we are able to predict the FACS measurements on never-before-seen patients, based on their clinical measurement alone. However, this relied upon the patients in the training set being representative of the patients in the held-out set. In practical applications, this means that the population of patients would need to be chosen carefully and diversely for the predictions to be meaningful for future patients.
The FMGAN model allows us to predict hard-to-obtain information for samples where we only directly measure EI. We demonstrate that the FMGAN can accurately model never-before-seen samples in these contexts. In the drug discovery context, this allows the potential impact of saving on expense and time by not performing as many physical experiments and instead modeling their results. In the clinical context, this allows for the modeling of patient data sooner, with more time to take positive interventions.

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

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

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

**DISCUSSION**

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

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

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

$$\text{min}_{G} \text{max}_{D} \mathbb{E}_{z \sim \mathcal{P}_{z}} \log (D(G(z))) + \mathbb{E}_{l \sim \mathcal{P}_{l}} \log (1 - D(G(z|l)))$$

Learning a generative model conditioned on the labels allows information sharing across labels, another advantage of the cGAN framework. Since the generator G must share weights across labels, the signal for any particular label l is blended with the signal from all other labels l ≠ i, allowing for learning without massive amounts of data for each label.

**Experimental Procedures**

**Resource availability**

**Lead contact**

The lead contact is Smita Krishnaswamy (smita.krishnaswamy@yale.edu).

**Materials availability**

There are no newly generated materials.

**GitHub**

https://github.com/KrishnaswamyLab/FMGAN.

**Conditional generative adversarial networks**

In a GAN, samples from the generator G can be obtained by taking samples from z ~ Z and then performing the forward pass with the learned weights of the network. But while the values of z control which points G generates, we do not know how to ask for specific types of points from G (more discussion of the original, unconditional GAN is in the supplemental information).

**Table 3. MMD scores on FACS measurements in clinical data experiments**

| COVID-19 FACS K-fold validation | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 |
|-------------------------------|--------|--------|--------|--------|--------|
| FMGAN                         | 0.039 ± 0.02 | 0.024 ± 0.01 | 0.022 ± 0.01 | 0.041 ± 0.02 | 0.031 ± 0.01 |
| Linear                        | 0.851 ± 0.03 | 0.915 ± 0.01 | 0.701 ± 0.02 | 0.758 ± 0.01 | 0.881 ± 0.01 |
| VAE                           | 0.623 ± 0.02 | 0.499 ± 0.01 | 0.682 ± 0.01 | 0.521 ± 0.01 | 0.588 ± 0.02 |

MMD distance between real and generated data (lower is better) on the COVID-19 data, with mean and standard deviation across the 26 held-out patients in each fold in the cross-evaluation. The FMGAN outperforms the baselines significantly in all cases.

**Chemical structure and SMILES strings**

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

**SMILES strings**

An SMILES string encodes the chemical structure of a drug in a variable-length set of standard letters and symbols. Each character in the string represents an element of the chemical’s physical formation, for example an atom, a bond, or a ring. For example, the common molecule glucose has the following structure: OC[C@@H](O)C[C@H](O)C[C@H](O)C[C@@H](O)1.

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

**Structure diagram**

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

**FMGAN architecture**

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

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

This input representation is then passed through $E$, a CNN, which produces the sequence embeddings. $E$ performs one-dimensional convolutions over each sequence followed by fully connected layers, eventually outputting a single $D_x$-dimensional vector for each SMILES string. We let these embeddings form the condition space for the next stage in FMGAN, the conditional $G$.

For the structure diagram experiment, we start with images that are points in $h \times w \times c$ space, with $h = w = 64$ and $c = 3$. They are then processed with a CNN. The CNN consists of four convolutional layers with stride 2, kernel size 3, and filters of 32, 64, 128, and 256, respectively. Batch normalization and a ReLU activation was used for each layer. Finally, after the convolutions, one fully connected layer maps the flattened output to a 100-dimensional point, representing the embedding learned for the particular diagram.

For both experiments, the generator structure, after the drugs are processed into conditions, is the same. Let $c_i$ be the condition for drug $i$ by the condition-embedding network. Let $x_i$ be the $D_x$-dimensional corresponding gene expression profile from a perturbation experiment performed with drug $i$. We build a GAN that trains a generator $G$ to model the underlying data distribution conditioned upon the structure $P_{\text{data}}(x|c)$. $G$ takes as input both a sample from a noise distribution (we choose an isotropic Gaussian) $z \sim N(0, I)$, and a condition $c$. $G$ maps these inputs to a $D_x$-dimensional point. Then, the discriminator $D$ takes both a $D_x$-dimensional point and a condition $c$ and outputs a single scalar representing whether it thinks the point was generated by $G$ or was a sample from $P_{\text{data}}$. These networks then train in the standard alternating gradient descent paradigm of GANs previously detailed.

For specific hyperparameter choices and data dimensionality details, we refer to the supplemental information.

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

**SUPPLEMENTAL INFORMATION**

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2021.100288.

**ACKNOWLEDGMENTS**

This research was supported, in part, by NIH grants R01GM135929 and R01GM130847 (to S.K.), Chan-Zuckerberg Initiative grants 182702 and CZF2019-002440 (to S.K.) and Novo Nordisk Interdisciplinary Synergy Program 2020 grant NNF20OC0063436 (to S.K.), and Project grants R01DK113191 and P30DK079310 (to P.W.).

**AUTHOR CONTRIBUTIONS**

M.A. and S.K. designed the model and experiments and wrote the text of the manuscript. M.A. performed the computational experiments and wrote the code. D.S. processed and advised on use of clinical dataset. D.B. processed and advised on use of drug dataset. D.D. helped come up with early ideas. P.W., M.S., Y.Y., D.D., F.W., and A.I. performed clinical experiments that gathered the clinical dataset.

**DECLARATION OF INTERESTS**

The authors declare no competing interests.

Received: August 19, 2020  
Revised: January 19, 2021  
Accepted: May 19, 2021  
Published: June 17, 2021

**REFERENCES**

1. Subramanian, A., Narayan, R., Corsello, S.M., Peck, D.D., Natoli, T.E., Lu, X., Gould, J., Davis, J.F., Tubelli, A.A., Asiedu, J.K., et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. Cell 177, 1437–1452.

2. Haider, S., and Pal, R. (2013). Inference of tumor inhibition pathways from drug perturbation data. In 2013 IEEE Global Conference on Signal and Information Processing (IEEE), pp. 95–98.

3. Larsson, O., Morita, M., Topisirovic, I., Alain, T., Blouin, M.J., Pollak, M., and Sonenberg, N. (2012). Distinct perturbation of the translumome by the anti-diabetic drug metformin. Proc. Natl. Acad. Sci. U S A 109, 8977–8982.

4. Korkut, A., Wang, W., Demir, E., Aksoy, B.A., Jing, X., Molinelli, E.J., Babur, Ö., Bernis, D.L., Onur Sumer, S., Solit, D.B., et al. (2015). Perturbation biology nominates upstream-downstream drug combinations in raf inhibitor resistant melanoma cells. Elife 4, e04640.

5. Lucas, C., Wang, P., Klein, J., Castro, T.B.R., Silva, J., Sundaram, M., Ellingson, M.K., Mao, T., Oh, J.E., Israelow, B., et al. (2020). Longitudinal analyses reveal immunological misfiring in severe covid-19. Nature 584, 463–469.

6. Krizhevsky A. and Hinton G.E. 2009. Learning multiple layers of features from tiny images.

7. Wang, X., Ghasedi Dizaji, K., and Huang, H. (2018). Conditional generative adversarial network for gene expression inference. Bioinformatics 34, i603–i611.

8. Xu, Y., Zhang, Z., You, L., Liu, J., Fan, Z., and Zhou, X. (2020), scigans: single-cell RNA-seq imputation using generative adversarial networks. Nucleic Acids Res. 48, e85.

9. Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D.S., Krebs, C.F., and Bonn, S. (2020). Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. Nat. Commun. 11, 1–12.

10. Moon, K.R., Stanley, J.S., Burkhardt, D., van Dijk, D., Wolf, G., and Krishnaswamy, S. (2018). 2. Manifold learning-based methods for analyzing single-cell RNA-sequencing data. Curr. Opin. Syst. Biol. 7, 36–46. https://doi.org/10.1016/j.coisbi.2017.12.008.

11. Gorban, A.N., Zinovyev, A., Gaussev, E., Cao, L., Gallinarri, P., Kwoj, J., Pasi, G., and Zaiane, O. (2015). Fast and user-friendly non-linear principal manifold learning by method of elastic maps. In 2015 IEEE International
12. Amodio, M., Van Dijk, D., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., et al. (2019). Exploring single-cell data with deep multitasking neural networks. Nat. Methods 16, 1–7.

13. Sidarta-Oliveira, D., and Velloso, L. (2020). Comprehensive Visualization of High-Dimensional Single-Cell Data with Diffusion-Based Manifold Approximation and Projection (Dbmap). Cell Reports. Published online May 17, 2020. https://doi.org/10.2139/ssrn.3582067.

14. McInnes, L., Healy, J., and Melville, J. (2018). Umap: uniform manifold approximation and projection for dimension reduction. arXiv, 1802.03426 https://arxiv.org/abs/1802.03426.

15. Dziugaite, G.K., Roy, D.M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. arXiv, 1505.03906 https://arxiv.org/abs/1505.03906.

16. Amodio, M., and Krishnaswamy, S. (2018). MAGAN: aligning biological manifolds. arXiv, 1803.00385 https://arxiv.org/abs/1803.00385.

17. Moon, K.R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D.B., Chen, W.S., Yim, K., van den Elzen, A., Him, M.J., Coifman, R.R., et al. (2019). Visualizing structure and transitions in high-dimensional biological data. Nat. Biotechnol. 37, 1482–1492.

18. Haimovich, A.D., Ravindra, N.G., Stoytchev, S., Young, H.P., Wilson, F.P., van Dijk, D., Schulz, W.L., and Taylor, R.A. (2020). Development and validation of the covid-19 severity index (csi): a prognostic tool for early respiratory decompensation. Ann Emerg Med. 76, 442–453.

19. Springenberg, J.T. (2015). Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv, 1511.06390 https://arxiv.org/abs/1511.06390.

20. Lotter, W., Kreiman, G., and Cox, D. (2015). Unsupervised learning of visual structure using predictive generative networks. arXiv, 1511.06380 https://arxiv.org/abs/1511.06380.