
Hierarchically branched diffusion models for scientific discovery

Alex M. Tseng¹ Max Shen¹ Tommaso Biancalani¹ Gabriele Scalia¹

Abstract

Although popular, current diffusion models have several drawbacks which limit their usefulness for biological discovery. Class-labeled datasets, such as those common in scientific domains, are rife with internal structure. Current class-conditional diffusion models, however, implicitly model diffusion on all classes in a flat fashion, ignoring any known relationships between classes. To leverage this structure, we propose hierarchically branched diffusion models as a novel framework for class-conditional generation. We highlight several advantages of branched diffusion models for scientific discovery: branched models are easily extended to novel classes in a continual-learning setting, they enable more sophisticated forms of conditional generation, and they offer a novel interpretability into the conditional-generation process. We extensively evaluate branched diffusion models on several benchmark and large real-world biological datasets, including a real-world single-cell RNA-seq dataset, where our branched model leverages the intrinsic hierarchical structure between human cell types.

1. Introduction

Diffusion models have gained major popularity as a method for generating data from complex data distributions, including when conditioning on labels (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021; Rombach et al., 2022). Despite these successes, however, current diffusion models are still limited in many scientific applications. Conventional diffusion models learn the diffusion process flatly for each label, disregarding any known relationships or structure between them. In reality, scientific data is typically characterized by a set of highly structured classes which can be thought of as hierarchical. For exam-

ple, human cell types are organized hierarchically by nature: keratinocytes are very distinct from neurons, but the latter subdivide into excitatory and inhibitory neurons.

In order to leverage this intrinsic structure, we propose restructuring diffusion models to be *hierarchically branched*, where the branching structure reflects the inherent relationships between distinct classes (the underlying diffusion process remains unchanged). This constitutes a novel way to perform class-conditional generation via diffusion. We apply branched diffusion to several benchmark and large real-world scientific datasets, including single-cell RNA-seq data and drug-like molecules, and highlight the following advantages of branched diffusion models for scientific discovery:

- They are easily extended to generate new, never-before-seen data classes in an efficient and principled manner (i.e. without requiring retraining the whole model). This is a critical requirement in scientific settings, as scientific datasets—such as single-cell atlases (Lotfolahi et al., 2021)—often grow steadily as data of new, never-before-seen classes (e.g. cell types, tissues, perturbations, etc.) is experimentally produced (Han et al., 2020; Almanzar et al., 2020).
- They allow transmuting objects from one class into the analogous object of a different class. For example, given a model trained on multiple cell types, with each cell type measured in various conditions or perturbations, we can use transmutation to answer questions like the following: “*what would be the expression of this specific B-cell lymphocyte treated with drug X, if the cell type were a T-cell lymphocyte instead?*”
- They offer interpretability into the underlying dataset, such as revealing common features between classes. Interpretability is a cornerstone of AI for science, but until now there has been limited work (if any) that attempts to improve and leverage diffusion-model interpretability.
- They are flexibly applied to many data types, any forward-diffusion scheme, and can be orthogonally combined with other methods that improve diffusion-model generative performance or efficiency.

¹Department of Artificial Intelligence and Machine Learning, Research and Early Development, Genentech. Correspondence to: Alex M. Tseng <tseng.alex@gene.com>.

2. Hierarchically branched diffusion models

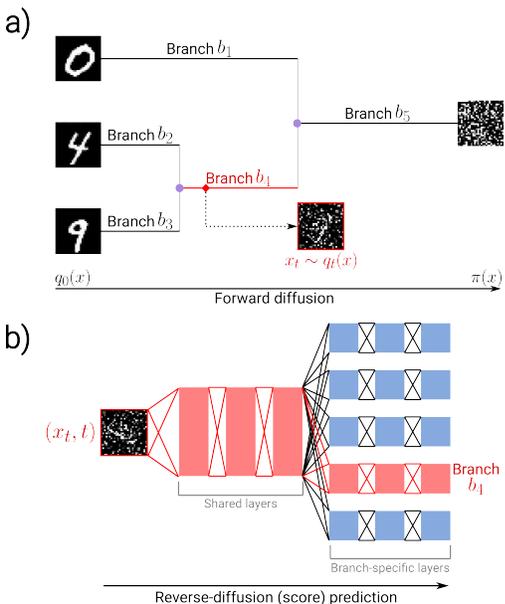


Figure 1. **a)** Illustration of branch points (purple dots) between classes. **b)** Each branch of a branched diffusion model is learned by a different task of a multi-task neural network.

Suppose our dataset consists of a set of classes C . Some classes in C are more similar than others (e.g. 4s and 9s in MNIST are visually more similar to each other than they are to 0s). As noise is progressively added to data, there is a point in diffusion time at which any two samples from two different classes are so noisy that their original class cannot be determined; we call this point in time a *branch point*. A branch point is a property of two classes (and the forward diffusion process) and—importantly—the more similar the two classes are, the earlier the branch point will be.

The branch points between all classes in C naturally encode a *hierarchy* of class similarities (Figure 1a). This hierarchy separates the diffusion from a single linear track into a branched structure, where each branch represents the diffusion of a subset of classes, and a subset of diffusion times. For $|C|$ classes, there are $2|C| - 1$ branches. Each branch $b_i = (s_i, t_i, C_i)$ of the diffusion model is defined by a particular diffusion time interval $[s_i, t_i)$ (where $0 \leq s_i < t_i < T$) and a subset of classes $C_i \subseteq C$ (where $C_i \neq \emptyset$). The branches are constrained such that every class and time $(c, t) \in C \times [0, T)$ can be assigned to exactly one branch b_i such that $c \in C_i$ and $t \in [s_i, t_i)$. The branches form a rooted tree starting from $t = T$ to $t = 0$. Late branches (large t) are shared across many different classes, as these classes diffuse nearly identically at later times. Early branches (small t) are unique to smaller subsets of classes. The earliest branches are responsible for

generating only a single class.

Additionally, as opposed to a conventional (“linear”) diffusion model which learns to reverse diffuse all classes and times using a single-task neural network, a branched diffusion model is implemented as a *multi-task neural network*, where each output task predicts reverse diffusion for a single branch (e.g. in an SDE-based diffusion framework (Song et al., 2021), each prediction head learns the Stein score for a specific branch) (Figure 1b). The multi-task architecture allows the model to learn the reverse-diffusion process distinctly for each branch, while the shared parameters allow the network to learn shared representations across tasks without an explosion in model complexity. Training a branched diffusion model follows nearly the same procedure as with a standard linear model, except for each input, we only perform gradient descent on the associated branch (i.e. model output task) (Algorithm S1). To sample an object of class c , we perform reverse diffusion starting from time T and follow the appropriate branch down (Algorithm S2).

Importantly, the underlying diffusion process in a branched diffusion model is *identical* to that of a conventional linear model; a branched model is characterized by the explicit definition of branch points which *separate* the responsibility of reverse diffusing different subsets of classes and times into separate branches, where each branch is predicted by a different head of a multi-task neural network.

Branched diffusion models are a completely novel method of class-conditional diffusion. Instead of relying on external classifiers or labels as auxiliary neural-network inputs, a branched diffusion model generates data of a specific class simply by reverse diffusing down the appropriate branches.

We demonstrate branched diffusion models on several datasets of different data modalities: 1) MNIST handwritten-digit images (LeCun et al.); 2) a tabular dataset of several features for the 26 English letters in various fonts (Frey & Slate, 1991); 3) a real-world, large scientific dataset of single-cell RNA-seq, measuring the gene expression levels of many blood cell types in COVID-19 patients, influenza patients, and healthy donors (Lee et al., 2020); and 4) ZINC250K, a large dataset of 250K real drug-like molecules (Irwin et al., 2012). For each dataset, we computed branch points using the algorithm in Appendix B. We trained continuous-time branched diffusion models for all datasets and verified they were generating high-quality samples (Supplementary Figure S1–S2). We compared the generative performance of our branched diffusion models to label-guided (linear) diffusion models, which are the current state-of-the-art method for conditional generation via diffusion (Ho et al., 2021). Our label-guided models were trained on the same data using a similar architecture and capacity. We computed the Fréchet inception distance (FID) for each class, comparing the branched diffusion mod-

els and their linear label-guided counterparts. In general, the branched diffusion models achieved similar generative performance—or better—compared to the current state-of-the-art label-guided strategy (Supplementary Figure S3). This establishes that branched diffusion models offer competitive performance in terms of sample quality. We also found that branched models remain robust to variation in the underlying branch points (Supplementary Figure S4).

3. Extending to novel classes

By separating the diffusion of different classes into distinct branches and output tasks of a neural network, a branched diffusion model easily accommodates the addition of new training data (e.g. from a recent experiment). This requirement is typical of large-scale, integrated scientific datasets, which are continuously updated with new research (e.g. single-cell reference atlases such as the Human Cell Atlas (Regev et al., 2017)). Leveraging the intrinsic structure of cell types (Han et al., 2020), a branched diffusion model can be fine-tuned on a new study—potentially containing new cell types—without retraining the entire model.

To illustrate this extendability, we trained a branched diffusion model on three MNIST classes: 0s, 4s, and 9s. We then introduced a new digit class: 7. In order to accommodate this new class, we added a single new branch to the diffusion model (Figure 2a). We then fine-tuned *only the newly added branch*, freezing all shared parameters and parameters for other output tasks. That is, we only trained on 7s, and only on times $t \in [s_i, t_i)$ for the newly added branch b_i . After fine-tuning, our branched diffusion model was capable of generating high-quality 7s *without affecting the ability to generate the other digits* (Figure 2b).

In contrast, if we start with a label-guided (linear) diffusion model (also trained on 0s, 4s, and 9s), it is much more difficult to extend the model to accommodate a new digit class. After fine-tuning the label-guided model on 7s, the model suffered from *catastrophic forgetting* (van de Ven & Tolias, 2019): it largely lost the ability to generate the other digits (even though the label of other digits was being fed to the model during sample generation), and generated almost all 7s for any label (Figure 2b). In order for the linear model to retain its ability to generate pre-existing digits, it must be retrained on the *entire* dataset, which is far more inefficient, particularly when the number of classes is large. In our MNIST example, retraining the linear model on all data took 7 times longer than training the singular new branch on a branched model. Notably, even after retraining on all classes, the linear model’s generation of old tasks still experienced inappropriate influence from the new task.

To quantify the ability of branched models to be extended to new data, we computed the FID of branched and label-

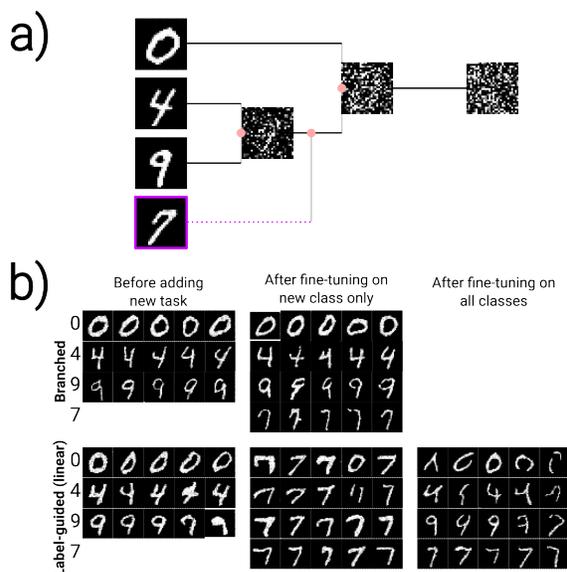


Figure 2. **a)** A trained branched diffusion model can accommodate a new class by adding and training a single new branch (purple dotted line). **b)** Examples of MNIST digits before and after introduction of new data class to branched versus linear model.

guided models before and after fine-tuning (Supplementary Figure S5). On both MNIST and the real-world single-cell RNA-seq dataset, we found that the branched model achieved roughly the same FIDs on pre-existing classes after fine-tuning on the new class. In contrast, fine-tuning the label-guided model on the new class caused the FID of other classes to become much worse. The label-guided model needed to be trained on the entire dataset to recover the FIDs of pre-existing classes, although the FID was still generally worse than the branched model.

4. Analogy-based conditional generation

In a diffusion model, we can traverse the diffusion process both forward and in reverse. Because branched models encode shared or interpolated characteristics between classes at branch points, this allows for a unique ability to perform *transmutation* (or *analogous* conditional generation) between classes. In transmutation, we start with an object from one class, forward diffuse to a branch point, and then reverse diffuse to a *different* class. This generates the analogous, corresponding object of a different class.

On our MNIST branched diffusion model, we transmuted between 4s and 9s (Figure 3a). Intriguingly, the model learned to transmute based on the *slantedness* of a digit. That is, slanted 4s tended to transmute to slanted 9s, and *vice versa*. To quantify the analogous conditional generation between classes, we transmuted between letters on our

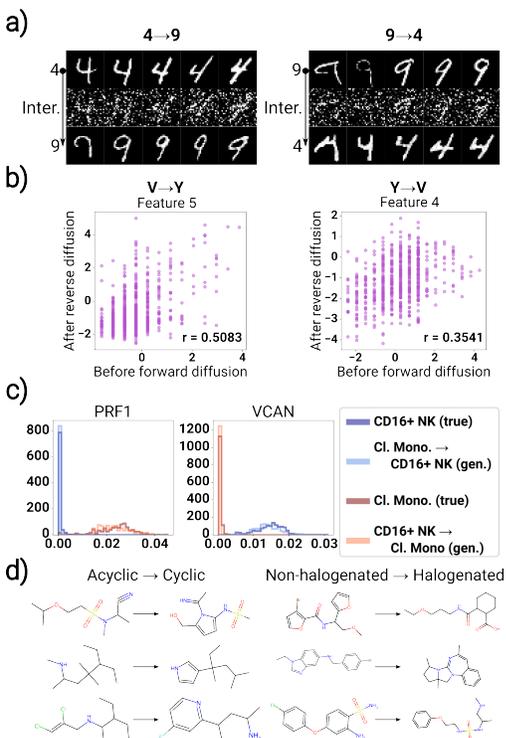


Figure 3. **a)** Examples of 4s transmuted to 9s (left), and 9s transmuted to 4s (right). **b)** Scatterplots of feature values before and after transmutation from Vs to Ys (left), or Ys to Vs (right). **c)** Distribution of critical marker genes before and after transmutation between CD16+ NK cells and classical monocytes. **d)** Examples of molecules we transmuted from acyclic to cyclic, and from non-halogenated and halogenated.

tabular branched diffusion model (Figure 3b). Transmuting between V and Y (and *vice versa*), we found that for every feature, there was a positive correlation of the feature values before versus after transmutation, *even if the feature range is different between the two classes*. This underscores the ability of branched diffusion models to transmute objects of one class to the *analogous* object of a different class.

On our real-world single-cell RNA-seq dataset, we transmuted a random sample of CD16+ NK cells to classical monocytes (and *vice versa*). In both directions, we found that the transmutation successfully increased critical marker genes of the target cell type, and zeroed out the marker genes of the source cell type (Figure 3c). Additionally, we found a high correlation in the expression of many genes throughout transmutation, including CXCL10 ($r = 0.20$), HLA-DRA ($r = 0.16$), and HLA-DRB1 ($r = 0.15$). These genes were explicitly identified and showcased in Lee et al. (2020) as being key inflammation genes that distinguish cells infected with COVID-19 from healthy cells. This illustrates how our branched model successfully transmuted COVID-infected

cells of one type into COVID-infected cells of another type (and reflexively, healthy cells from one type into healthy cells of another type). Finally, we trained a branched diffusion model on ZINC250K. We were able to transmute molecules into analogous molecules of desired properties, while largely retaining core functional groups (e.g. amines, esters, sulfonamides, etc.) (Figure 3d).

5. Interpretable hybrids at branch points

In the reverse-diffusion process, branch points are where distinct classes split off and begin reverse diffusing along different trajectories. Thus, for two similar classes (or two sets of classes), the reverse-diffusion intermediate at a branch point naturally encodes features which are shared (or otherwise intermediate or interpolated) between the two.

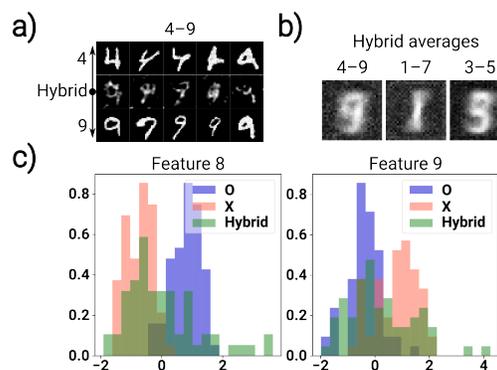


Figure 4. **a)** Examples of MNIST hybrids between the digits classes 4 and 9. Each hybrid in the middle row is the reverse-diffusion starting point for both images above and below it. **b)** Aggregate hybrids at branch points (averaged over many samples) between pairs of MNIST classes. **c)** Distribution of feature values between two pairs of letter classes—O and X—and in the generated hybrids from the corresponding branch point.

For example, on our MNIST branched diffusion model, these “hybrids” at branch points tend to show shared characteristics that underpin both digit distributions (Figure 4a–b). On our branched model trained on tabular letters, we see that hybrids tend to interpolate between distinct feature distributions underpinning the two classes, acting as a smooth transition state between the two endpoints (Figure 4c).

6. Conclusion

We proposed branched diffusion models as an alternative method of class-conditional generation. This novel framework holds several advantages for datasets with highly structured classes—such as those in scientific settings—which we demonstrated on several benchmark and large-scale real-world datasets.

References

- Almanzar, N., Antony, J., Baghel, A. S., Bakerman, I., Bansal, I., and et. al. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature* 2020 583:7817, 583:590–595, 7 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2496-1. URL <https://www.nature.com/articles/s41586-020-2496-1>.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 12 2021.
- Frey, P. W. and Slate, D. J. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6:161–182, 1991. ISSN 15730565. doi: 10.1023/A:1022606404104/METRICS. URL <https://link.springer.com/article/10.1023/A:1022606404104>.
- Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., Chen, H., Wang, J., Tang, H., Ge, W., Zhou, Y., Ye, F., Jiang, M., Wu, J., Xiao, Y., Jia, X., Zhang, T., Ma, X., Zhang, Q., Bai, X., Lai, S., Yu, C., Zhu, L., Lin, R., Gao, Y., Wang, M., Wu, Y., Zhang, J., Zhan, R., Zhu, S., Hu, H., Wang, C., Chen, M., Huang, H., Liang, T., Chen, J., Wang, W., Zhang, D., and Guo, G. Construction of a human cell landscape at single-cell level. *Nature* 2020 581:7808, 581:303–309, 3 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2157-4. URL <https://www.nature.com/articles/s41586-020-2157-4>.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. URL <https://github.com/hojonathanho/diffusion>.
- Ho, J., Research, G., and Salimans, T. Classifier-free diffusion guidance, 11 2021.
- Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52:1757–1768, 7 2012. ISSN 1549960X. doi: 10.1021/CI3001277/SUPPL_FILE/CI3001277_SI_001.PDF. URL <https://pubs.acs.org/doi/full/10.1021/ci3001277>.
- LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. URL <http://yann.lecun.com/exdb/mnist/>.
- Lee, J. S., Park, S., Jeong, H. W., Ahn, J. Y., Choi, S. J., Lee, H., Choi, B., Nam, S. K., Sa, M., Kwon, J. S., Jeong, S. J., Lee, H. K., Park, S. H., Park, S. H., Choi, J. Y., Kim, S. H., Jung, I., and Shin, E. C. Immunophenotyping of covid-19 and influenza highlights the role of type i interferons in development of severe covid-19. *Science Immunology*, 5:1554, 7 2020. ISSN 24709468. doi: 10.1126/SCIIMMUNOL.ABD1554/SUPPL_FILE/TABLE_S9.XLSX. URL <https://www.science.org/doi/10.1126/sciimmunol.abd1554>.
- Lotfollahi, M., Naghipourfar, M., Luecken, M. D., Khajavi, M., Büttner, M., Wagenstetter, M., Žiga Avsec, Gayoso, A., Yosef, N., Interlandi, M., Rybakov, S., Misharin, A. V., and Theis, F. J. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology* 2021 40:1, 40:121–130, 8 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-01001-7. URL <https://www.nature.com/articles/s41587-021-01001-7>.
- Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., and et. al. The human cell atlas. *eLife*, 6, 12 2017. ISSN 2050084X. doi: 10.7554/ELIFE.27041.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 6 2022. doi: 10.1109/CVPR52688.2022.01042. URL <https://ieeexplore.ieee.org/document/9878449/>.
- Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *32nd International Conference on Machine Learning, ICML 2015*, 3:2246–2255, 3 2015. doi: 10.48550/arxiv.1503.03585. URL <https://arxiv.org/abs/1503.03585v8>.
- Song, Y., Sohl-Dickstein, J., Brain, G., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. 2021.
- van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. 4 2019. doi: 10.48550/arxiv.1904.07734. URL <https://arxiv.org/abs/1904.07734v1>.

A. Supplementary Figures and Tables

Algorithm S1 Training a branched diffusion model

Input: training set $\{(x^{(k)}, c^{(k)})\}$, branches $\{b_i\}$
repeat
 Sample (x_0, c) from training data $\{(x^{(k)}, c^{(k)})\}$
 Sample $t \sim Unif(0, T)$
 Forward diffuse $x_t \sim q_t(x|x_0)$
 Find branch $b_i = (s_i, t_i, C_i)$ s.t. $s_i \leq t < t_i, c \in C_i$
 Gradient descent on $p_{(\theta_s, \theta_i)}(x_t, t)[i]$ (on output task i)
until convergence

Algorithm S2 Sampling a branched diffusion model

Input: class c , trained p_θ , branches $\{b_i\}$
Sample $\hat{x} \leftarrow x_T$ from $\pi(x)$
for $t = T$ to 0 **do**
 Find branch $b_i = (s_i, t_i, C_i)$ s.t. $s_i \leq t < t_i, c \in C_i$
 $\hat{x} \leftarrow p_\theta(\hat{x}, t)[i]$ (take output task i)
end for
Return \hat{x}

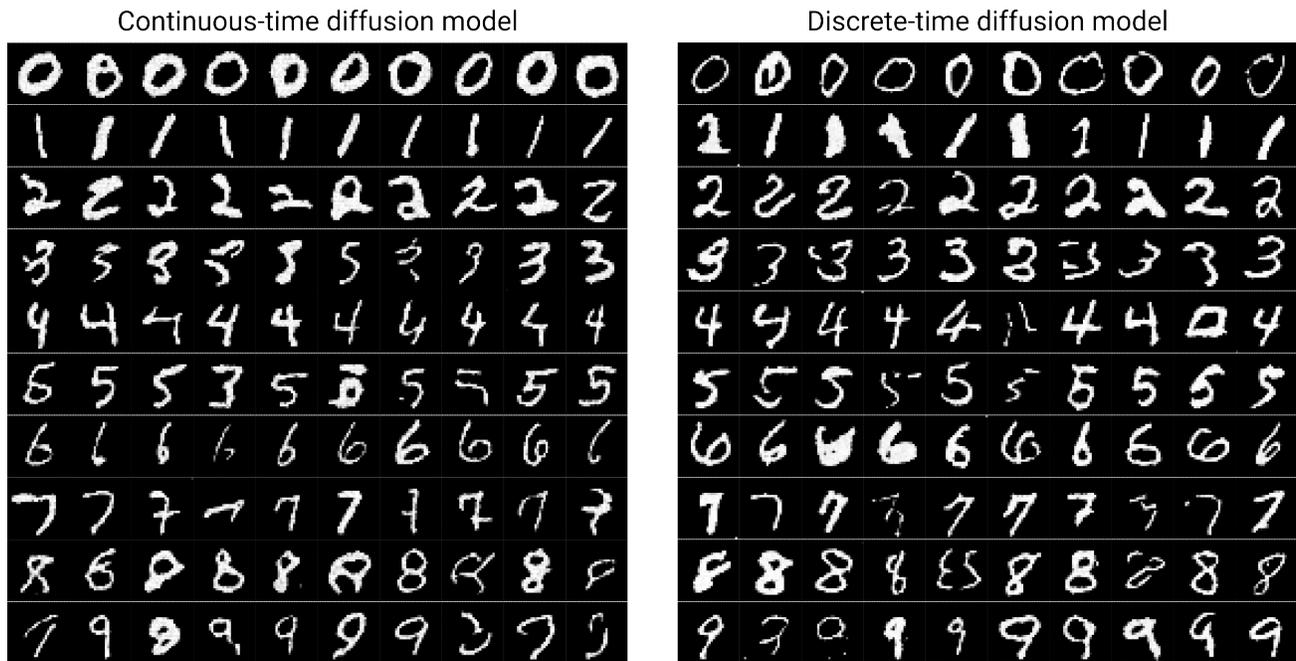


Figure S1. Examples of generated MNIST images. We show (uncurated) images of MNIST digits generated by branched diffusion models. Since branched diffusion models naturally output each class separately, generation of individual classes does not require supplying labels or pretrained classifiers. We show a sample of digits generated from a continuous-time (score-matching) diffusion model (Song et al., 2021), and a discrete-time diffusion model (denoising diffusion probabilistic model) (Ho et al., 2020). Branched diffusion models for multi-class generation fit neatly into practically any diffusion-model framework.

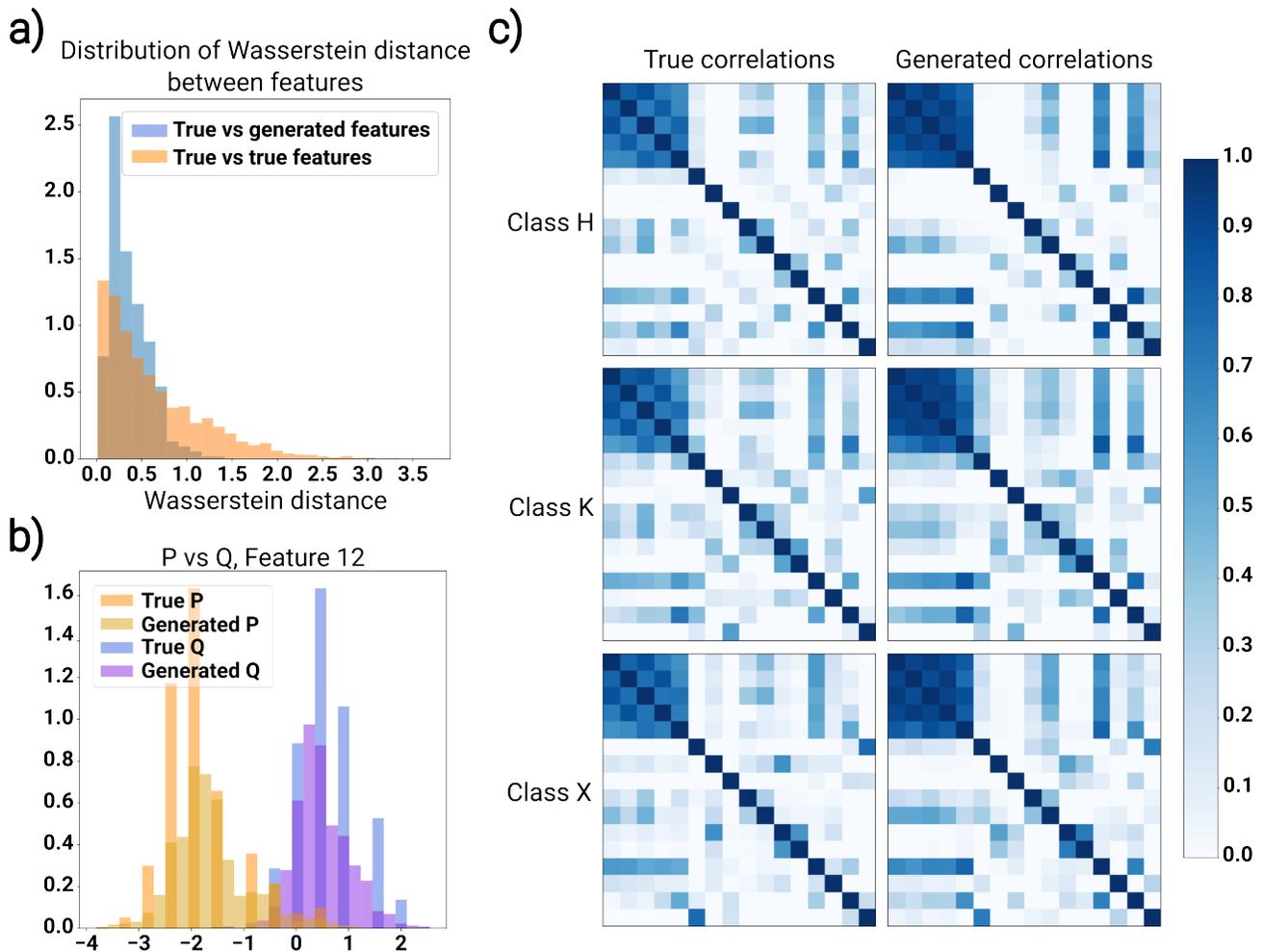


Figure S2. Examples of generated letters. We show some examples of distributions generated from a branched diffusion model trained on tabular data: English letters of various fonts, featurized by a hand-engineered set of 16 features. **a)** For each letter class and each of the 16 numerical features, we computed the Wasserstein distance (i.e. earthmover’s distance) between the true data distribution and the generated data distribution. We compare this distribution of Wasserstein distances to the distances between different true distributions of features as a baseline. On average, the branched diffusion model learned to generate features which are similar in distribution to the true data. **b)** We show an example of the true and generated feature distributions for a particular feature, comparing two letter classes: P and Q. Although the two classes show a very distinct distribution for this feature, the branched diffusion model captured this distinction well and correctly generated the feature distribution for each class. **c)** Over all 16 numerical features, we computed the Pearson correlation between the features, and compare the correlation heatmaps between the true data and the generated examples. In each of these three classes, the branched diffusion model learned to capture not only the overarching correlational structure shared by all three classes, but also the subtle secondary correlations unique to each class.

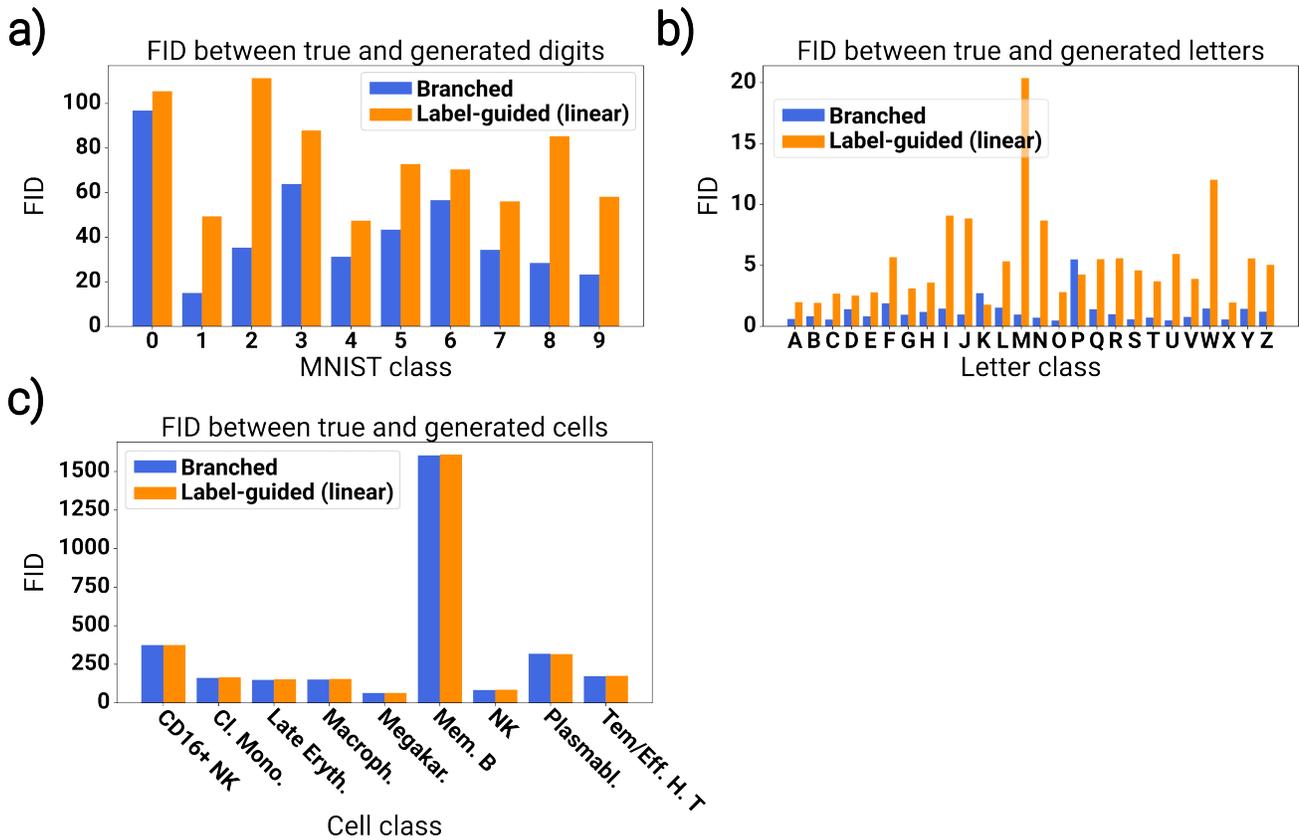


Figure S3. Sample quality of branched diffusion vs label-guided (linear) diffusion. We compare the quality of generated data from branched diffusion models to label-guided (linear) diffusion models of similar capacity and architecture. For each class, we computed the Fréchet inception distance (FID) between the generated examples and a sample of the true data. A lower FID is better. We show the FID for generated **a)** MNIST digits; **b)** tabular letters; and **c)** single-cell RNA-seq. We find that our branched diffusion model achieved comparable sample quality compared to the current state-of-the-art method of label-guided diffusion. In some cases, the branched model even consistently generated better examples.

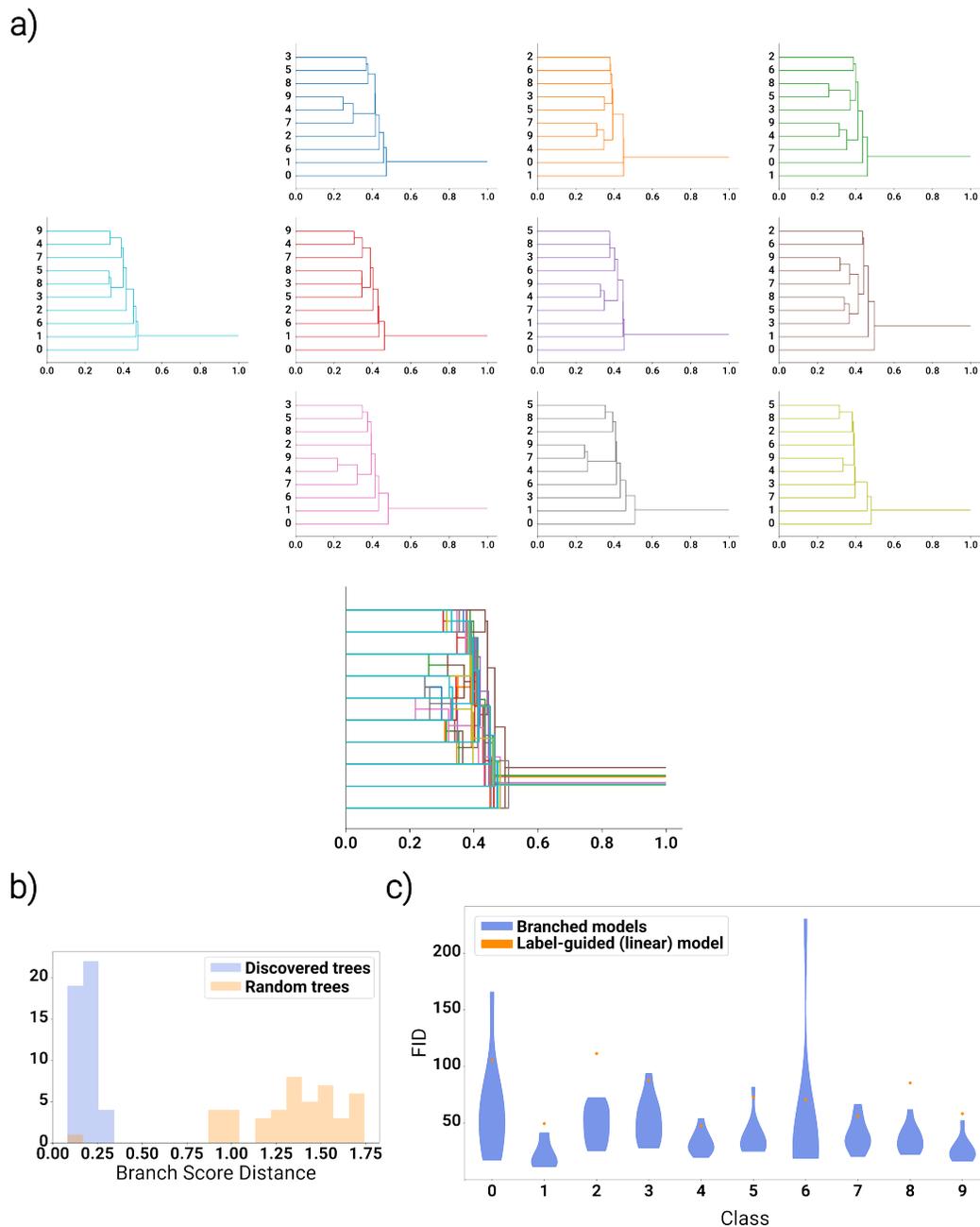


Figure S4. Robustness of branch points. a) We computed branch points and hierarchies for the MNIST dataset 10 times, each time resulting in a slightly different branching structure. The variation results from randomness in sampling from the dataset, and randomness from the forward-diffusion process. The 10 branching structures vary not only in their branching times, but in their topologies (above). To emphasize the variation in the hierarchies, we also overlay all 10 hierarchies on the same axes (below). **b)** Compared to randomly generated hierarchies, the branching structures generated by our algorithm (Supplementary Methods Section ??) have a much lower branch-score distance between themselves ($p < 10^{-25}$ by Wilcoxon test). **c)** We trained a branched diffusion model on each of the hierarchies, and quantified generative performance using Fréchet inception distance (FID). Over all 10 hierarchies, the FID from the branched models were relatively consistent with each other, and also generally better than the label-guided (linear) model.

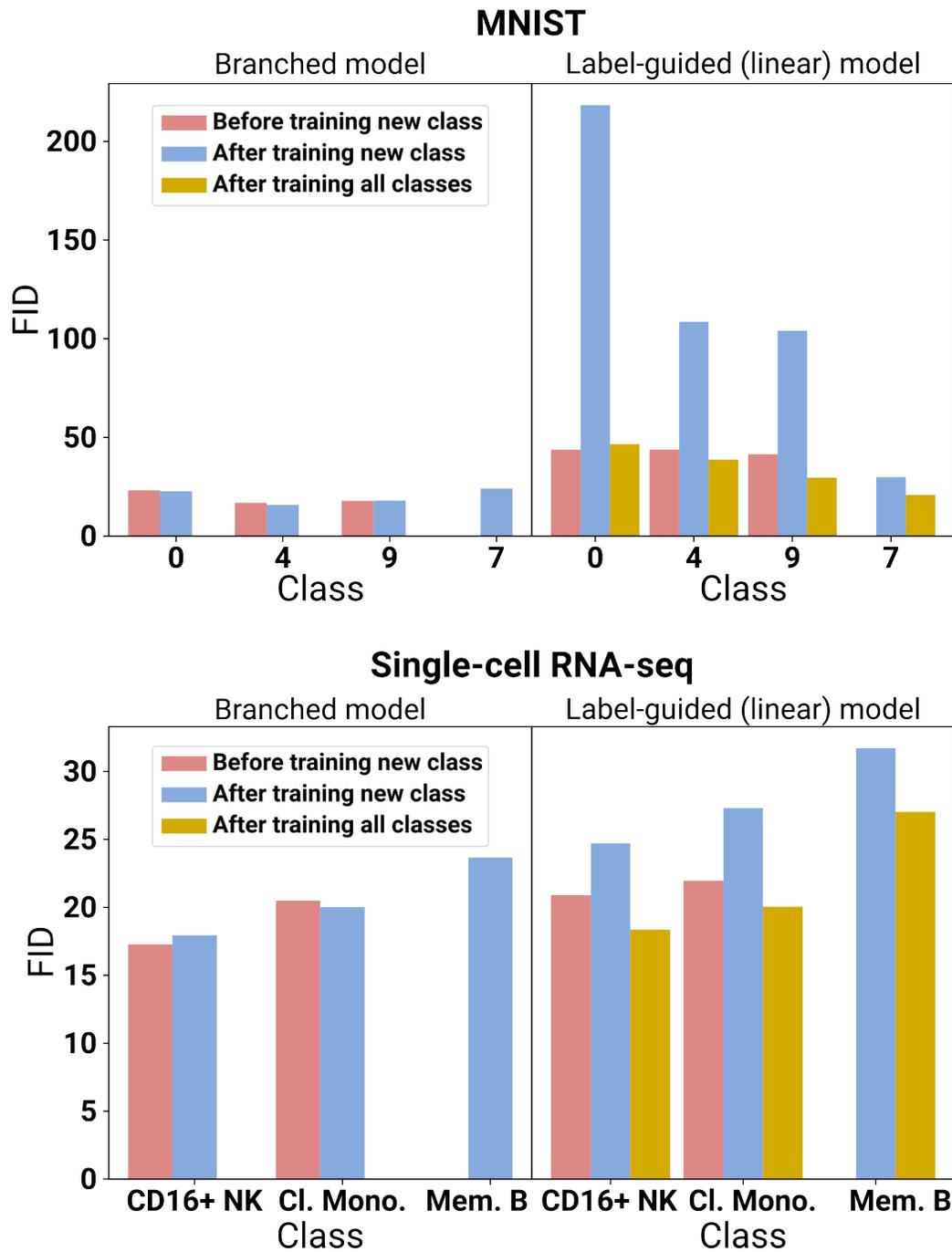


Figure S5. FID after class extension. In order to quantify the ability of branched models to be extended to new data classes, we computed the FID of branched and label-guided models before and after fine-tuning on a new, never-before-seen data class. On both MNIST (above) and the real-world single-cell RNA-seq dataset (below), we found that the branched model achieved roughly the same FIDs on pre-existing classes after fine-tuning on the new data class. In contrast, fine-tuning the label-guided model on the new data class caused the FID of other classes to significantly worsen. The label-guided model needed to be trained on the entire dataset to recover the FIDs of pre-existing classes, although the FIDs were still generally worse than those of the branched model.

B. Branch-point discovery algorithm

In a branched diffusion model, each branch $b_i = (s_i, t_i, C_i)$ learns to reverse diffuse between times $[s_i, t_i]$ for classes in C_i . The branches form a tree structure (i.e. hierarchy) with the root at time T and a branch for each individual class at time 0. These branch definitions may come from prior domain knowledge, or they can be computed from the training data alone. In our work, we computed the branch definitions using the following algorithm:

1. Start with a dataset of objects to generate, consisting of classes C .
2. For each class, sample n objects randomly and without replacement.
3. Forward diffuse each object over 1000 time points in the forward-diffusion process (we used 1000 steps, as this matched the number of reverse-diffusion steps we used for sample generation). The branched diffusion model which will be trained using these branch definitions employs an identical forward-diffusion process.
4. At each time point t , compute the average similarity of each pair of classes, resulting in a $|C| \times |C|$ similarity matrix at each of the 1000 time points. For distinct classes c_i, c_j ($i \neq j$), the similarity $s(t, c_i, c_j)$ is computed over the average of n pairs, where the pairs are randomly assigned between the two classes; for self-similarity of class c_i , the similarity $s(t, c_i, c_i)$ is computed over the average of n pairs within the class, randomly assigned such that the same object is not compared with itself. For simplicity, the similarity metric we used was Euclidean distance over the flattened vectors, but other metrics may be used which better match the domain.
5. For each pair of classes c_i, c_j (i may be equal to j), smooth the trajectory of $s(t, c_i, c_j)$ over time by applying a Gaussian smoothing kernel of standard deviation equal to 3 and truncated to 4 standard deviations on each side.
6. For each pair of *distinct* classes c_i, c_j ($i \neq j$), compute the *earliest* time in the forward-diffusion process such that the average similarity between c_i and c_j is at least the self-similarity of c_i and c_j (averaged between the two). A tolerance of ϵ is allowed. That is, for each pair of distinct classes c_i, c_j ($i \neq j$), compute the *minimum* t such that $s(t, c_i, c_j) \geq \frac{1}{2}(s(t, c_i, c_i) + s(t, c_j, c_j)) - \epsilon$. This gives each pair of distinct classes a “minimal time of indistinguishability”, τ_{c_i, c_j} .
7. Order the $\binom{|C|}{2}$ minimal times of indistinguishability τ by ascending order, and greedily build a hierarchical tree by merging classes together if they have not already been merged. This can be implemented by a set of $|C|$ disjoint sets, where each set contains one class; iterating through the times τ in order, two branches merge into a new branch by merging together the sets containing the two classes, unless they are already in the same set.