1. Create inhomogeneous Markov chain for each column.
   For each column $i$:
   you need initial probability $p^+_i$: probability the top pixel in the column is a 1.
   - estimate using the fraction of images where column $i$ starts with a 1.
   - transition probabilities for $i = 2:28$
     $p(x^+_i=1|x^-_{i-1}=1)$: probability pixel $i$ is a 1 given pixel $i-1$ is a 1.
     $p(x^+_i=0|x^-_{i-1}=0)$: probability pixel $i$ is a 1 given pixel $i-1$ is a 0.
     - again estimate using counts.

Can store transition probabilities in $28 \times 28 \times 2$ tensor:

$P_i(i,:,:)$  
$p(x^+_{i+1}|x^-_i = 1)$

When sampling:

- if $x^-_{i-1} = 1$
  - sample $x^+_i$ according to $P_i(i,1,1)$
- if $x^-_{i-1} = 0$
  - sample $x^+_i$ according to $P_i(i,1,2)$
2. For each (i, t) in each image we create a DAG

Let's call these the parents $P_1, \ldots, P_n$

For each pixel (i, t) get the set of parents in each image

and the value of pixel (i, t) in each image

Models $f_1, f_2$ = binary Tabular ($\tilde{x}_{it}, y_{it}, x = i$)

When sampling,

for each (i, t)

get values & parents

$p = [P_1, P_2, \ldots, P_n]$

$m = \text{models } \tilde{x}_{it}, f_1$

$I(i, t) = m \cdot \text{sample}(m, p)$
4. Almost identical to 2.
- set & parents changes
- use different encoding for $y$

$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ instead of } \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- sampler returns \{-1 or 1\} - convert back to \{0 or 1\} when setting pixel values