Table 1: Tile types in Mario levels. Symbol characters come from the VGLC. Identity values are used for one-hot encoding. Visualizations are used by the Mario AI framework.

Tile type	Symbol	Identity	Visualization
Empty/Sky (passable)	-	0	
Top-left pipe	<	1	
Top-right pipe	>	2	
Full question block	?	3	
Cannon top	B	4	
Enemy	E	5	
Empty question block	Q	6	
Breakable	S	7	
Solid/Ground	X	8	
Left pipe	[	9	
Right pipe	]	10	
Cannon support	b	11	
Coin	o	12	

Appendix

Additional hyperparameter settings and results that are only in the arXiv pre-print.

Dataset Details

Our cleaned version of the VGLC and our captioning approach resulted in data with the following properties:

- Number of Super Mario Bros. levels: 20
- Number of Super Mario Bros. 2 levels: 22
- Total 16×16 samples across both games: 7,687
- Vocabulary size for `regular` captions: 47
- Vocabulary size for `absence` captions: 48
- Training samples: 6918
- Validation samples: 384
- Test samples: 385

The tiles available in Mario levels are in Table 1.

Text Encoder Details

These details are relevant to our `MLM` model:

- Token embedding size: 128
- Number of transformer encoder layers: 4
- Number of attention heads: 8
- Dimension of hidden layer: 256
- Probability of [MASK] token during MLM training: 0.15
- Training optimizer: AdamW
- Training epochs: 300
- Loss function: Cross Entropy Loss
- Learning rate: Starts at 0.00005
- Minimum learning rate: 0.000001
- Learning rate schedule: ReduceLROnPlateau
- Training batch size 16

Diffusion Model Details

These details are relevant to our diffusion models:

- Base dimension of the UNet: 128
- Number of residual blocks for downsampling: 2
- UNet encoder (down) channels: 13, 128, 256, 512
- UNet decoder (up) channels: 512, 256, 128, 13
- Number of attention heads: 8
- Noise schedule: DDPM with a linear beta schedule
- Noise betas: 0.0001 to 0.02
- Noise schedule time steps: up to 1000
- Training optimizer: AdamW
- AdamW weight decay: 0.01
- AdamW beta values: 0.9 and 0.999
- Gradient accumulation steps: 1
- Learning rate schedule: cosine
- Learning rate warm-up period: 25 epochs
- Top learning rate: 0.0001
- Guidance scale during inference: 7.5
- Inference steps: 30

The loss function for the diffusion model is the same one used by Lee and Simo-Serra (2023):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda \mathcal{L}_{\text{rec}} \tag{5}$$

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\epsilon}_i - \epsilon_i\|^2 \tag{6}$$

$$\mathcal{L}_{\text{rec}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} \log P_\theta(O_{i,h,w} | x_{i,h,w}) \tag{7}$$

where $\lambda = 0.001$ is the weight on the reconstruction loss, N is the batch size, $\hat{\epsilon}_i$ is the model's predicted noise for sample i, ϵ_i is the true noise, H and W are the height and width of 16, $O_{i,h,w}$ is the ground truth for the tile at position (h, w) in sample i, $x_{i,h,w}$ is the generated tile at position (h, w) in sample i, so $P_\theta(O_{i,h,w}|x_{i,h,w})$ is the probability of the original block given the generated block according to the diffusion model with parameters θ.

Figure 7 depicts the complete diffusion pipeline for training and inference.

Five-Dollar Model Details

These details are relevant to our Five-Dollar Models:

- Number of residual blocks: 3
- Number of convolutional filters: 128
- Kernel size: 7, but 3 for final layer
- Noise vector size: 5
- Training epochs: 100
- Loss function: Negative Log Likelihood Loss
- Training optimizer: AdamW
- Learning rate: 0.001

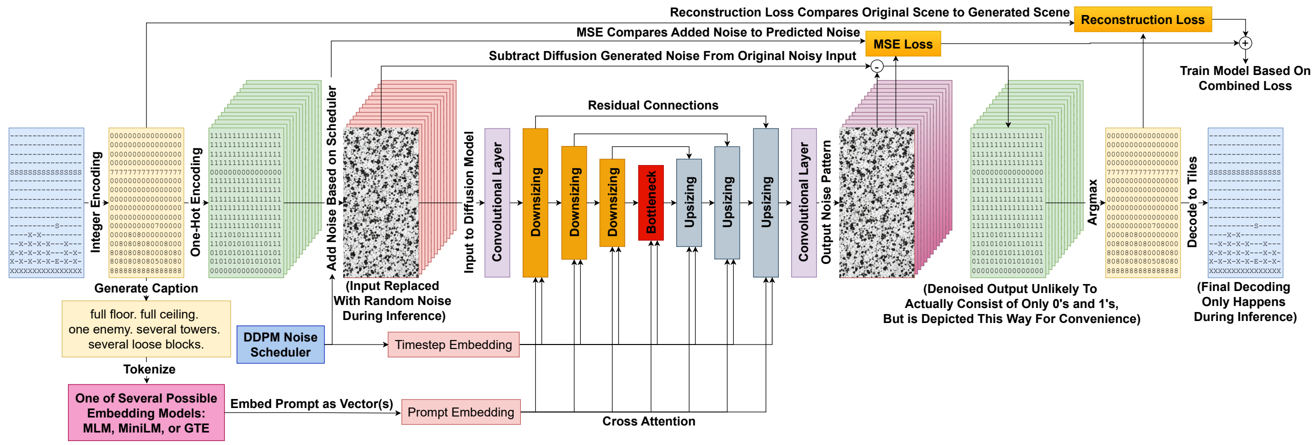
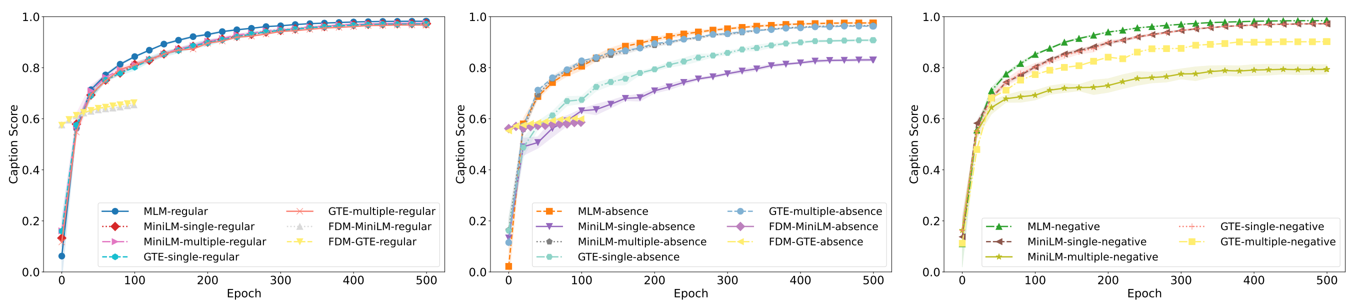


Figure 7: Diffusion Training and Inference Pipeline. Our training set is integer-encoded, but was derived from the ASCII data in the VGLC. Each scene is associated with an automatically generated caption. The scenes are one-hot encoded before noise is added according to a DDPM scheduler. The noisy input enters the diffusion model, while its cross-attention blocks access a hidden state based on both a timestep embedding from DDPM and a prompt embedding from whichever text model is being used. The output of the model is a noise prediction which is directly compared to the known amount of added noise to complete the Mean Squared Error. The predicted noise is also removed from the noisy input to approximate the one-hot encoded training data, which is then integer-encoded via argmax for comparison to the original training sample. This is how the reconstruction loss is calculated. Both losses are combined to train the model.



(a) `regular` (b) `absence` (c) `negative`

Figure 8: Average Caption Adherence Score by Epoch on All Real Game Captions. Results are qualitatively similar to those in Figure 2. (a) `regular` caption results. (b) `absence` caption results. (c) `negative` caption results.

Additional Performance Metrics and Results

Results dealing with these performance metrics could not fit into the main text of the paper.

Caption Adherence on Full Dataset When applied to the set of all captions from the original games (Figure 8), the caption adherence score is qualitatively similar to the results from just the test set data, as demonstrated earlier (Figure 2).

End Time and Best Time on Logarithmic Scale Most execution times are small, but a few larger values skew the presentation in Figure 4a. The same data from that figure is depicted in Figure 9 using a logarithmic scale.

Caption Order Tolerance We want to give users the flexibility to provide caption phrases in whatever order they prefer. Semantically, a caption is equivalent to any caption that is a permutation of its phrases. We can take a caption and sample some number of its permutations, send each one through a text-to-level model, and average the c-scores:

$$\text{tolerance}(P) = \frac{\sum_{(p,c)\in P} \text{c-score}(p,c)}{|P|} \tag{8}$$

P is a set of pairs (p, c), where p is a prompt and c is the caption on the level a model produces using p. Values of p are distinct permutations of the same input prompt.

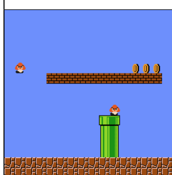
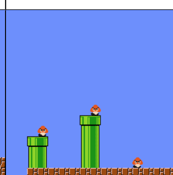
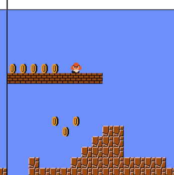
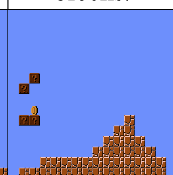
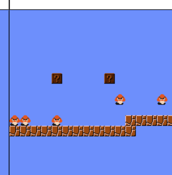
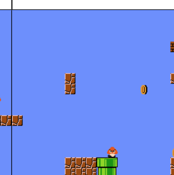
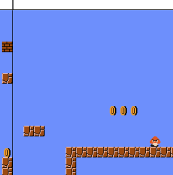
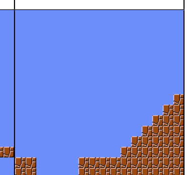
Prompts can contain many phrases, so averaging across all permutations would be computationally expensive. Instead, we sample up to 5 distinct random permutations per prompt.

Caption order tolerance results are in Figure 10.

Making Larger Levels

Tables 2, 3, and 4 show different examples of using the interactive GUI to create longer levels.

Table 2: Long Level Generated One Scene At a Time (16 wide). Using `MLM-regular0` (https://huggingface.co/schrum2/MarioDiffusion-MLM-regular0), the GUI was used to generate 16×16 scenes with the designated prompts. The segments were then combined into a single playable level. The caption adherence score of each scene is shown beneath it.

full floor. one platform. two enemies. one pipe. a few coins.	floor with one gap. a few enemies. a few pipes. one tower.	floor with several gaps. two platforms. one rectangular block cluster. a few enemies. many coins. one tower. one ascending staircase.	floor with several gaps. one rectangular block cluster. one irregular block cluster. several enemies. many coins. one tower. one ascending staircase. one descending staircase. two question blocks.	a few platforms. several enemies. one question block. two loose blocks.	giant gap with one chunk of floor. a few platforms. one rectangular block cluster. two enemies. one pipe. a few coins. a few loose blocks.	giant gap with one chunk of floor. a few platforms. a few enemies. a few coins. one coin line. one question block. a few loose blocks.	floor with one gap. one ascending staircase.
							
0.888889	0.875	0.347222	0.458333	0.972222	0.722222	0.819444	1.0

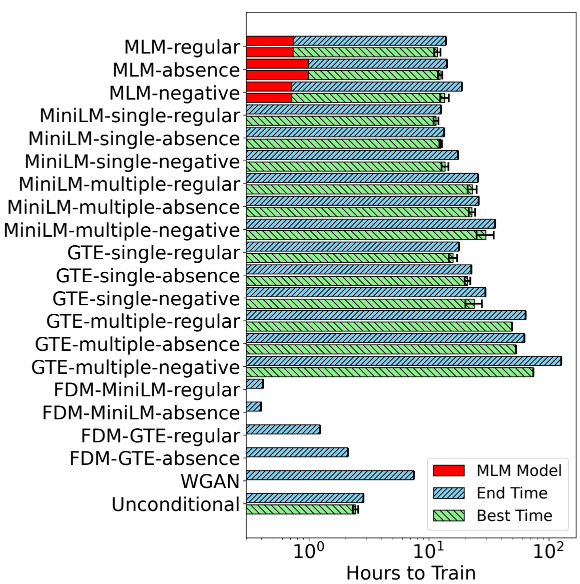


Figure 9: Average End Times and Best Times on Log Scale.

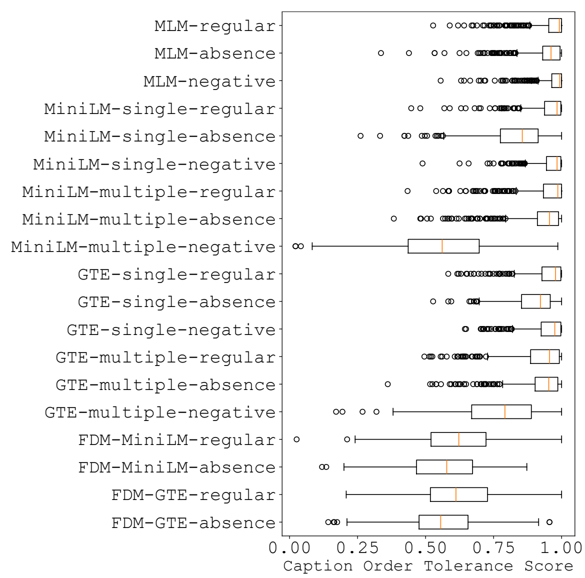


Figure 10: Caption Order Tolerance. Shows how models handle different phrase orderings in captions of real game scenes. One model of each type is considered with scores for each caption in the test set. Most models do well, except `MiniLM-single-absence`, `MiniLM-multiple-negative`, `GTE-multiple-negative`, and FDM.

Table 3: Long Level Generated One Scene At a Time (32 wide). Using `MLM-regular0` (https://huggingface.co/schrum2/MarioDiffusion-MLM-regular0), the GUI was used to generate 32×16 scenes with the designated prompts. The segments were then combined into a single playable level. Each segment of width 32 has its own caption adherence score, but the result of splitting each segment into two scenes of width 16 and averaging those caption adherence scores is also shown. It is generally harder to control the output and to interpret the meaning of the caption adherence scores when the width increases.

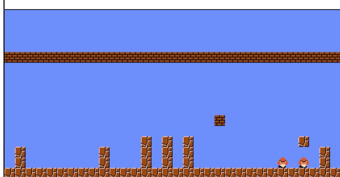
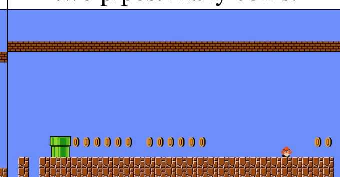
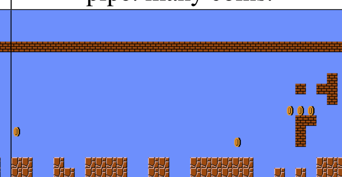
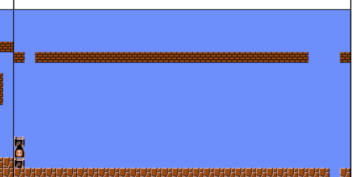
full floor. full ceiling. one enemy. many coins. one coin line. several towers.	floor with two gaps. ceiling with two gaps. one rectangular block cluster. one irregular block cluster. a few enemies. two pipes. many coins.	giant gap with several chunks of floor. ceiling with one gap. two irregular block clusters. one enemy. one upside down pipe. many coins.	floor with two gaps. ceiling with one gap. one platform. two cannons. a few question blocks.
			
0.63888889 AVG: 0.56944444	0.56111111 AVG: 0.46388889	0.38055556 AVG: 0.33888889	0.75 AVG: 0.69027778

Table 4: Long Level Generated One Scene At a Time (64 wide). Using `MLM-regular0` (https://huggingface.co/schrum2/MarioDiffusion-MLM-regular0), the GUI was used to generate 64×16 scenes with the designated prompts. The segments were then combined into a single playable level. Each segment of width 64 has its own caption adherence score, but the result of splitting each segment into four scenes of width 16 and averaging those caption adherence scores is also shown. Note that it is not necessary for level widths to be multiples of 16, nor is it necessary for all segments in a level to have the same width.

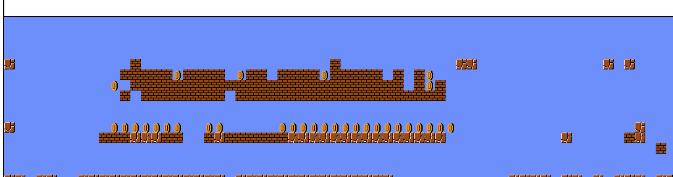
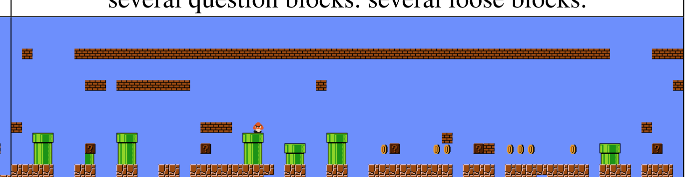
floor with several gaps. a few platforms. one rectangular block cluster. one irregular block cluster. a few enemies. a few coin lines. many coins. one ascending staircase. a few loose blocks.	floor with two gaps. ceiling with one gap. several platforms. a few rectangular block clusters. one irregular block cluster. a few enemies. two pipes. one coin line. two coins. one cannon. several question blocks. several loose blocks.
	
0.638889 AVG: 0.479167	0.458333 AVG: 0.280556

Table 5: Example scenes generated by models trained with regular captions. Each of these models is available on Hugging Face (Details here: https://github.com/schrum2/MarioDiffusion/blob/main/MODELS.md). The first row shows the prompt used to generate the scene. The first five columns are real captions from the test set, and the next five are from the random test set of captions not present in the original data. Beneath each image is the resulting caption adherence score. These images are also available online: https://people.southwestern.edu/~schrum2/mario.html.

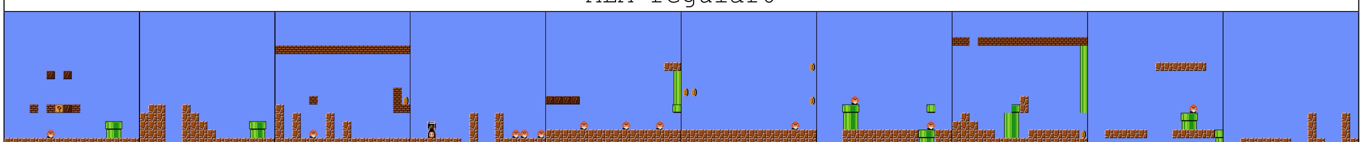
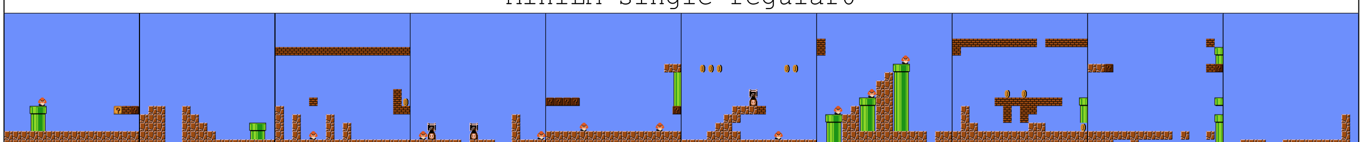
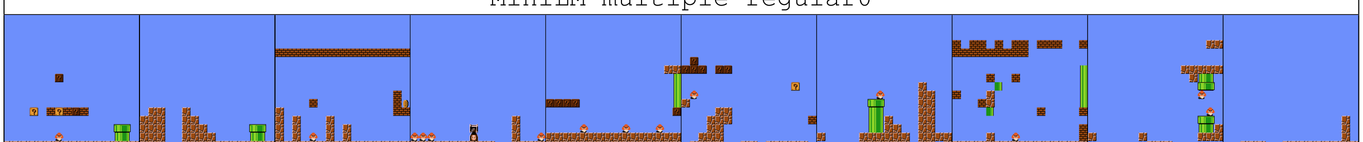
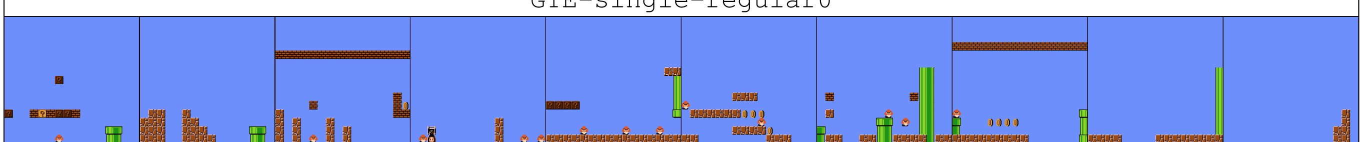
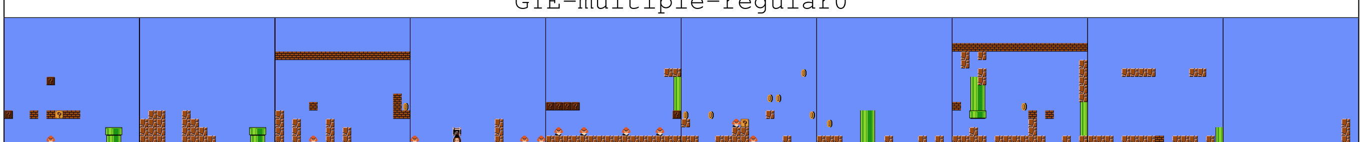
full floor. one enemy. a few question blocks. one platform. one pipe.	floor with one gap. one descending staircase. one pipe. one irregular block cluster.	full floor. full ceiling. one enemy. one coin. one irregular block cluster. a few towers. a few loose blocks.	floor with one gap. a few enemies. one cannon. one tower.	full floor. a few enemies. a few question blocks. one platform. one upside down pipe. two loose blocks.	a few coin lines. one irregular block cluster. a few enemies. several coins. two ascending staircases. one question block. one rectangular block cluster. two cannons.	floor with several gaps. two pipes. two enemies. one descending staircase. two towers. two upside down pipes.	full floor. one descending staircase. one loose block. a few upside down pipes. full ceiling. two coins. one enemy.	several platforms. two rectangular block clusters. one pipe. a few upside down pipes.	floor with several gaps. one tower.
MLM-regular0									
									
0.88888889	1.0	1.0	0.97222222	1.0	0.26388889	0.51388889	0.35277778	0.625	0.97222222
MiniLM-single-regular0									
									
0.98611111	1.0	1.0	0.95833333	0.75	0.48611111	0.38888889	0.13055556	0.5	0.98611111
MiniLM-multiple-regular0									
									
1.0	1.0	1.0	1.0	0.76388889	0.02777778	0.40277778	0.22777778	0.375	0.97222222
GTE-single-regular0									
									
0.98611111	1.0	1.0	1.0	1.0	0.27777778	0.31944444	0.38055556	0.5	0.98611111
GTE-multiple-regular0									
									
0.88888889	1.0	1.0	1.0	0.76388889	0.38888889	−0.0138889	0.375	0.33333333	0.98611111

Table 6: Example scenes generated by models trained with absence captions. Each of these models is available on Hugging Face (Details here: https://github.com/schrum2/MarioDiffusion/blob/main/MODELS.md). The first row shows the regular prompt that the actual input prompt is based on. Phrases for absent concepts are added automatically. The first five columns are real captions from the test set, and the next five are from the random test set of captions not present in the original data. Beneath each image is the resulting caption adherence score. These images are also available online: https://people.southwestern.edu/~schrum2/mario.html.

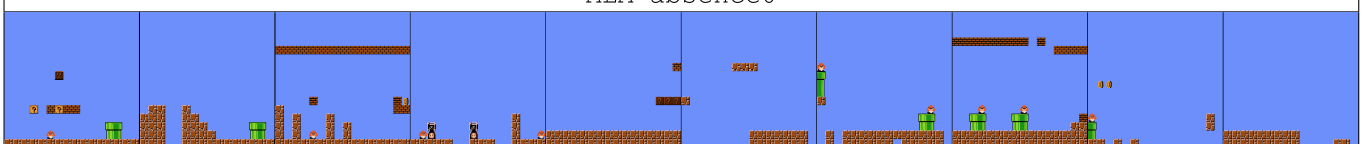
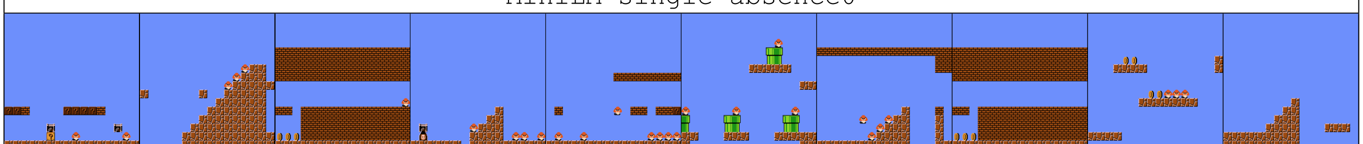
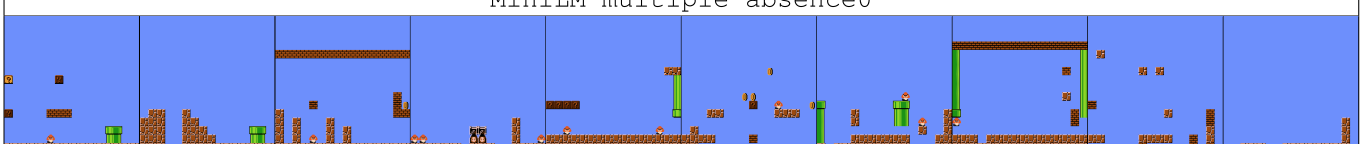
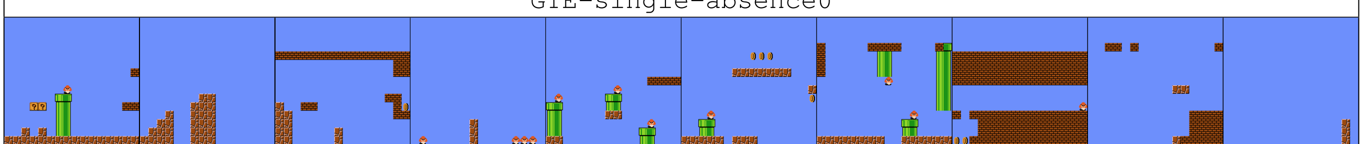
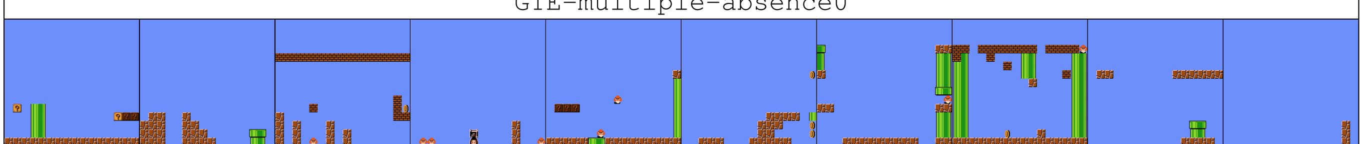
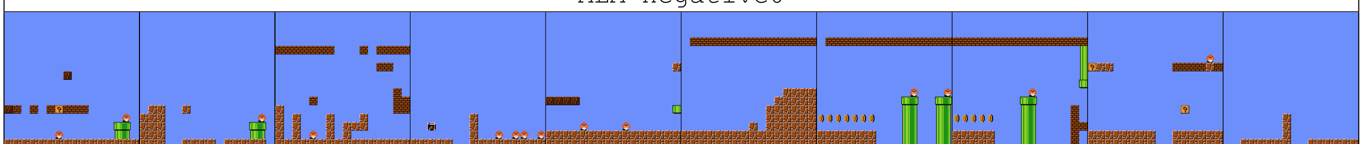
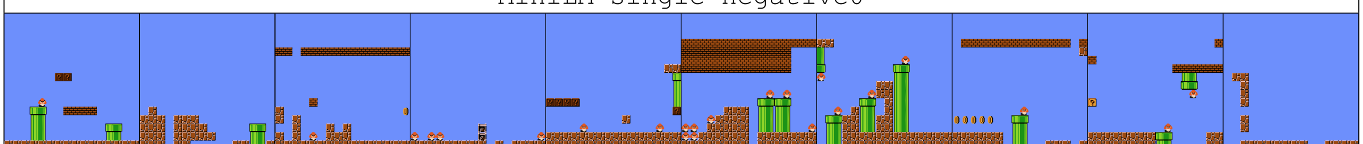
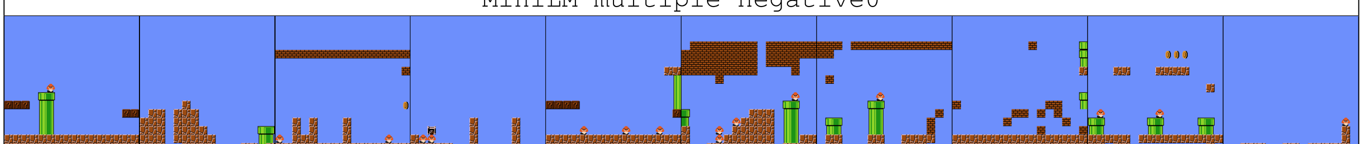
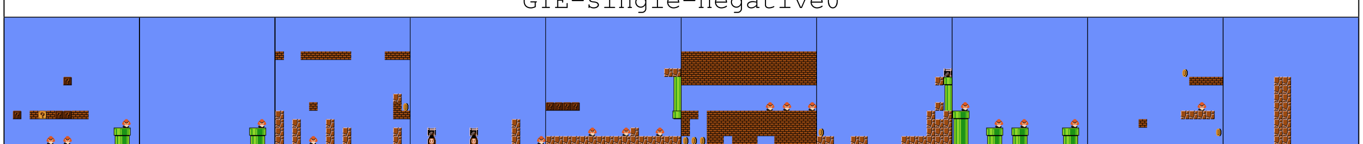
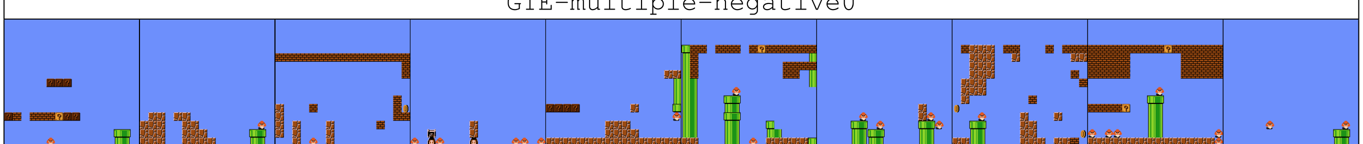
full floor. one enemy. a few question blocks. one platform. one pipe.	floor with one gap. one descending staircase. one pipe. one irregular block cluster.	full floor. full ceiling. one enemy. one coin. one irregular block cluster. a few towers. a few loose blocks.	floor with one gap. a few enemies. one cannon. one tower.	full floor. a few enemies. a few question blocks. one platform. one upside down pipe. two loose blocks.	a few coin lines. one irregular block cluster. a few enemies. several coins. two ascending staircases. one question block. one rectangular block cluster. two cannons.	floor with several gaps. two pipes. two enemies. one descending staircase. two towers. two upside down pipes.	full floor. one descending staircase. one loose block. a few upside down pipes. full ceiling. two coins. one enemy.	several platforms. two rectangular block clusters. one pipe. a few upside down pipes.	floor with several gaps. one tower.
`MLM-absence0`									
									
1.0	1.0	1.0	0.95833333	0.76388889	−0.2222222	0.43055556	0.15833333	0.11111111	0.75
`MiniLM-single-absence0`									
									
0.73611111	0.22222222	0.30555556	0.77777778	0.80555556	−0.1388889	0.26388889	0.20833333	0.20833333	0.625
`MiniLM-multiple-absence0`									
									
1.0	1.0	1.0	0.97222222	0.98611111	0.26388889	0.51388889	0.5	0.30555556	0.98611111
`GTE-single-absence0`									
									
0.86111111	0.66666667	0.625	0.88888889	0.65277778	0.15277778	0.27777778	0.44444444	0.52777778	0.98611111
`GTE-multiple-absence0`									
									
0.66666667	1.0	1.0	1.0	0.63888889	0.09722222	0.26388889	0.56111111	0.73611111	0.98611111

Table 7: Example scenes generated by models trained with negative captions. Each of these models is available on Hugging Face (Details here: https://github.com/schrum2/MarioDiffusion/blob/main/MODELS.md). The first row shows the regular prompt. Phrases for absent concepts automatically create the corresponding negative prompt. The first five columns are real captions from the test set, and the next five are from the random test set of captions not present in the original data. Beneath each image is the resulting caption adherence score. These images are also available online: https://people.southwestern.edu/~schrum2/mario.html.

full floor. one enemy. a few question blocks. one platform. one pipe.	floor with one gap. one descending staircase. one pipe. one irregular block cluster.	full floor. full ceiling. one enemy. one coin. one irregular block cluster. a few towers. a few loose blocks.	floor with one gap. a few enemies. one cannon. one tower.	full floor. a few enemies. a few question blocks. one platform. one upside down pipe. two loose blocks.	a few coin lines. one irregular block cluster. a few enemies. several coins. two ascending staircases. one question block. one rectangular block cluster. two cannons.	floor with several gaps. two pipes. two enemies. one descending staircase. two towers. two upside down pipes.	full floor. one descending staircase. one loose block. a few upside down pipes. full ceiling. two coins. one enemy.	several platforms. two rectangular block clusters. one pipe. a few upside down pipes.	floor with several gaps. one tower.
MLM-negative0									
									
0.86111111	0.63888889	0.68611111	1.0	0.75	−0.0138889	0.30555556	0.33888889	0.30555556	0.97222222
MiniLM-single-negative0									
									
0.95833333	0.875	0.92222222	0.86111111	0.73611111	−0.0416667	0.56944444	0.31666667	0.47222222	0.63888889
MiniLM-multiple-negative0									
									
0.97222222	0.63888889	0.93055556	0.83333333	0.76388889	−0.0416667	0.30555556	0.06944444	0.27777778	0.65277778
GTE-single-negative0									
									
0.95833333	0.63888889	0.93611111	0.94444444	0.98611111	0.38888889	0.06944444	0.30555556	0.19444444	0.95833333
GTE-multiple-negative0									
									
0.95833333	0.76388889	0.875	0.86111111	0.73611111	−0.2361111	0.40277778	0.30277778	0.38888889	0.65277778