Weekly Study Report

Wang Ma

2025-02-11

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute

Outline

Towards Understanding and Quantifying Uncertainty for Text-to-Image Generation

1. Towards Understanding and Quantifying Uncertainty for Text-to-Image Generation

1.1 Uncertainty Quantification in Vanilla Diffusion Models (unconditional)

DDPM-OOD: determine whether a test data x^* is from the training p_{data}.

1. Add noise to x^* in forward process to get a sequence of noisy images $\left\{x^*_{t_1}, x^*_{t_2}, \ldots x^*_T\right\}$
2. Denoise the sequence of noisy images to get reconstructions $\{\hat{x}^*_1, \hat{x}^*_2, \ldots\}$

Reconstruction Error. $\mathrm{Error} = \dfrac{1}{N}\sum_i^N \|x^* - \hat{x}^*_i\|$

Uncertainty Approach. $\mathrm{Unc} = \mathrm{var}(\hat{x}^*)$

Similarly, the above approach can be done in Latent Space (LMD

Given a conditional model $p_\theta(x|c)$ which models the conditional data distribution $p_{\text{data}}(x|c)$. Conditioned on a prompt c^*, we draw x^* from $p_\theta(x^*|c^*)$. In this case, we wish to quantify the uncertainty of model p_θ with respect to the condition c^*. In this case, uncertainties should be concerned with semantics between text prompt and the generated output image.

- Aleatoric Uncertainty: non-reducible. High aleatoric uncertainty would be where a variation of generated concepts may arise from a single prompt. For example, a spelling mistake where *fish* is mistyped as *fis*, then the model may result in a *fist* in the image.
- Epistemic Uncertainty: should correspond to a models' lack of knowledge of the semantic concepts in the prompt. For example, a model trained on ImageNet won't know what the president Trump looks like, thus will have high epistemic uncertainty for this semantic concept.

1.3 Prompt-based Uncertainty Estimation

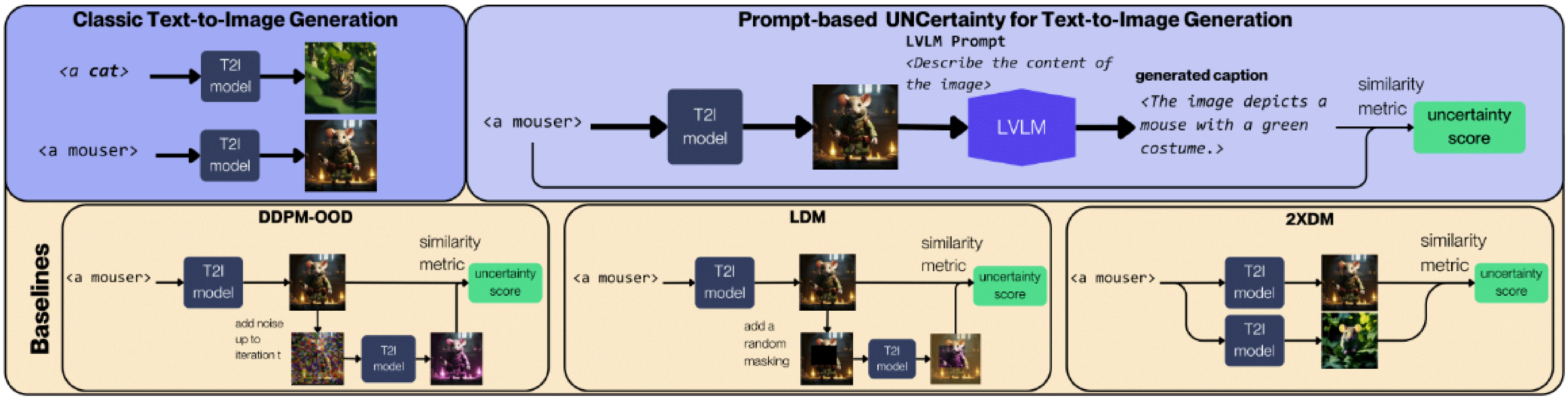


Figure 4. **Illustration showing the different baselines and PUNC.** PUNC leverages a LVLM to describe generated images and assess similarity with the original prompt, providing a refined uncertainty score. In contrast, baseline methods employ traditional techniques such as noise injection, ensembling, or masking to quantify uncertainty, followed by image-based similarity scoring.

Background on Large Vision-Language Model (LVLM).

Given a text prompt c and an image x, a LVLM model with parameters θ, includes

- Image Encoder: $f_\theta^{\text{img}}(\cdot)$ processes the input image x and produces an embedding $z^{\text{img}} = f_\theta^{\text{img}}(x)$
- Text Processor with LLM: $f_\theta^{\text{txt}}(\cdot, \cdot)$ takes the prompt c and the image embedding z^{img} as inputs, generating a descriptive or answer-based response: $\hat{c} = f_\theta^{\text{txt}}(c, z^{\text{img}})$

Thus, a LVLM can generate an interpretation $\hat{c}$ that reflects the content of the image given an initial prompt.

Workflows.

- Step 1: Initial Prompt and Image Generation. Given a test prompt c^*, we use the T2I model to sample an image x:

$$x \sim p_\theta(x|c^*)$$

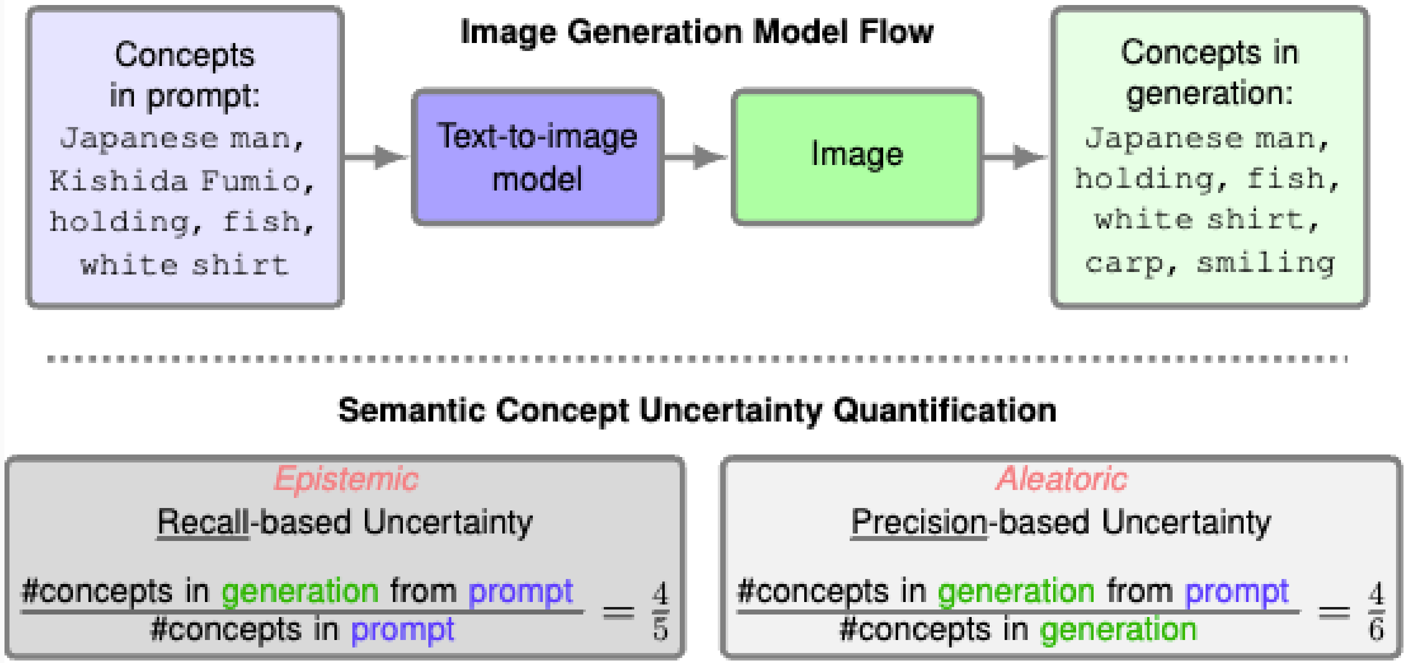
- Step 2: Image Interpretation via LVLM. With the generated image x and the initial prompt c^*, the LVLM produces a new descriptive caption $\hat{c}$:

$$\hat{c} = f_\theta^{\text{txt}}\left(c, f_\theta^{\text{img}}(x)\right)$$

- Step 3: Uncertainty Score Calculation. Using the similarity score $S(c^*, \hat{c})$:

$$S(c^*, \hat{c}) = \text{sim}(c^*, \hat{c})$$

High Similarity $\rightarrow$ Low Epistemic Uncertainty



Intuitively, a *lack of knowledge* about concepts in the prompt will result in fewer concepts being repserved in the image, reducing *recall*.

A *lack of specificity* in the prompt will result in additional concepts being generated, reducing *precision*.

Reducing LLM Hallucinations using Epistemic Neural Networks

2. Reducing LLM Hallucinations using Epistemic Neural Networks

1. Chain of Thoughts (CoT)
2. Self-ensembling techniques: sample many reasoning chains and impose self-consistency to obtain a self-ensembled answer.
3. Retrieval-Augmented Generation (RAG): the model will use the query to search for a small number of highly relevant documents with factual information, then the LLM will be asked to generate a responce for the query given such auxiliary information.
4. Fine-tuning for specific tasks.

- A possible reason for a Language Model to hallucinate is the maximum likelihood language modelling objective, which minimize the forward KL divergence between the data and model distributions. This objective potentially results in a model with **mass-seeking** behavior which causes the LM to assign non-zero probability to sentences that are not fully consistent with knowledge embedded in the training data.
- Representation Difference in different layers in a Language Model:
  1. Early layers: encode “lower-level” information (e.g., part-of-speech tags)
  2. Later layers: more “semantic information” (more factual)

  So,
  1. Later logits mainly contain factual information and knowledge
  2. Early logits mainly contain meaningful but not necessarily true

2.2 DoLa (Decoding by Contrasting Layers)

Figure 1: Dynamic premature layer selection in DoLa

1. **Premature Layer Selection.** Different models have different best premature Layer. DoLa aims to select a best premature layer so that its output distribution is the most different from the final layer.

$$M = \arg\max_{j \in J} \mathrm{JSD}\ \left(q_F(\cdot \mid x_{<t}) \parallel q_j(\cdot \mid x_{<t})\right),$$

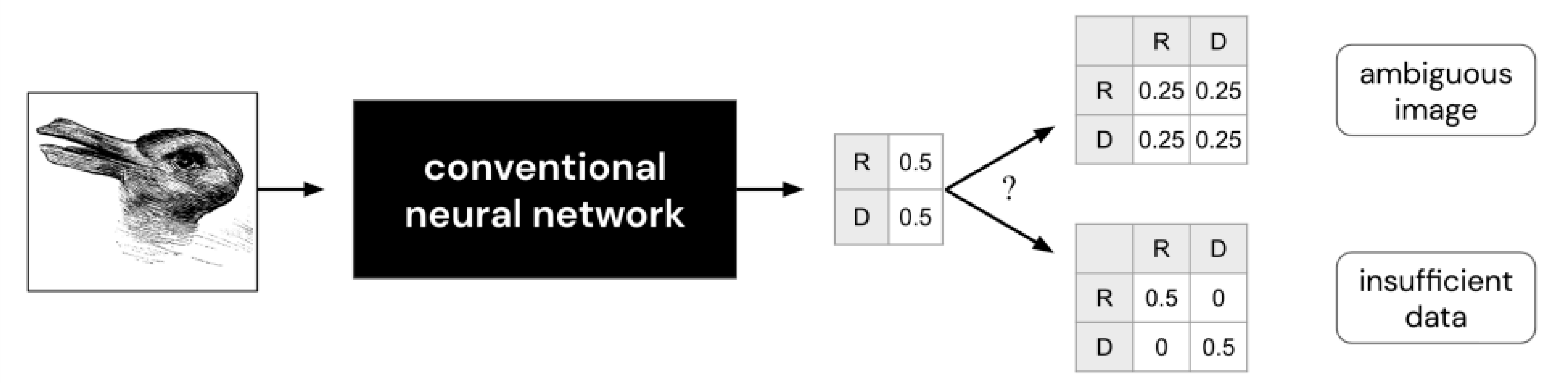
   where J is all candidates of premature layers, q_F is the final output distribution, q_j is the output distribution of j-th layer, and JSD is the Jensen-Shannon Divergence.

2. **Contrastive decoding approach**: amplify the output from the final layer while downplaying the output from the premature layer.

$$F(q_F(x_t), q_E(x_t)) := \begin{cases} \log \frac{q_F(x_t)}{q_E(x_t)} & \text{if } x_t \in V_{\text{head}}(x_t \mid x_{<t}) \\ -\infty & \text{othervise} \end{cases}$$

$$\hat{p}(x_t) = \mathrm{softmax}\ \left(F(q_F(x_t), q_E(x_t))\right)$$

Measuring Joint Distribution



If we want to get a joint distribution from traditional way:

$$\hat{P}_{\{1:2\}}(x) = \prod_{t=1}^{2} \text{softmax}\ (f_\theta(x))$$

, this assumes the two prediction result are independent, not the true joint distribution.

2.3 Epistemic Neural Networks

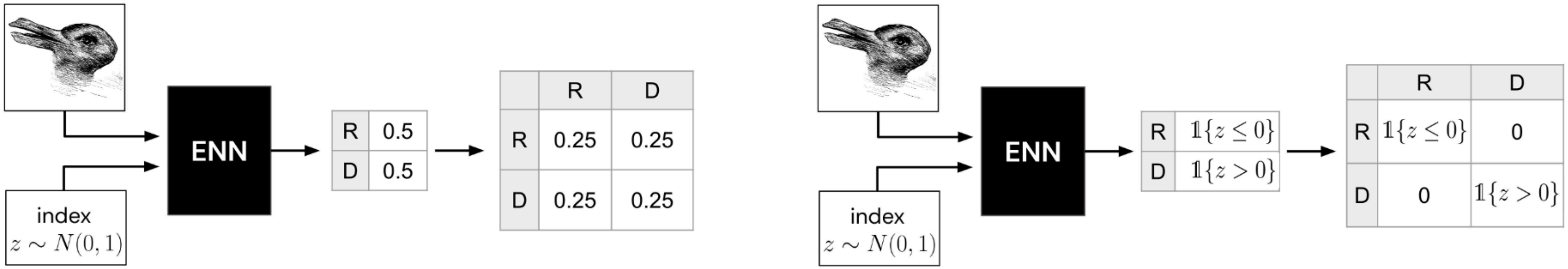
Epistemic NN way: $\hat{P}_{\{1:2\}}(x) = \int_z P_z(z) \prod_{t=1}^{2} \operatorname{softmax}\left(f_\theta(x, z)\right)$

z:

- can be understood as condition or assumption
- can be understood as a “domain information”
- can be some simple distribution, such as uniform distribution on finite sets or standard Gaussian.

Why z:

1. intuitively, different zs result in different prediction with same NN, which can be considered as an “ensemble”
2. If model does not have a good understanding of the test data, then given different z, the model will output completely different results (high epistemic uncertainty)
3. if the data is sufficiency, any zs will make similar prediction, which means the model has less “reducible” uncertainty.



(a) An ENN indicating an ambiguous image. (b) An ENN indicating insufficient data.

Figure 3: An ENN can incorporate the epistemic index $z \sim P_Z$ into its joint predictions. This allows an ENN to differentiate inevitable ambiguity from data insufficiency.

Algorithm 1 ENN training via SGD

Inputs:

dataset	training examples $\mathcal{D}=\{(x_i,y_i,i)\}_{i=1}^N$
ENN	network f, reference P_Z, initialization θ_0
loss	ℓ evaluates example (x_i,y_i,i) for index z
batch size	data samples n_B, index samples n_Z
optimizer	update rule and number of iterations T

Returns:

θ_T	parameter estimates for the ENN.

```
for t in 0,...,T-1 do
    sample data Ĩ = i_1,..,i_{n_B} ~ Unif({1,..,N}).
    sample indices Z̃ = z_1,..,z_{n_Z} ~ P_Z.
    compute grad ← ∇_{θ|θ=θ_t} Σ_{z∈Z̃} Σ_{i∈Ĩ} ℓ(θ,z,x_i,y_i,i).
    update θ_{t+1} ← optimizer(θ_t,grad)
```

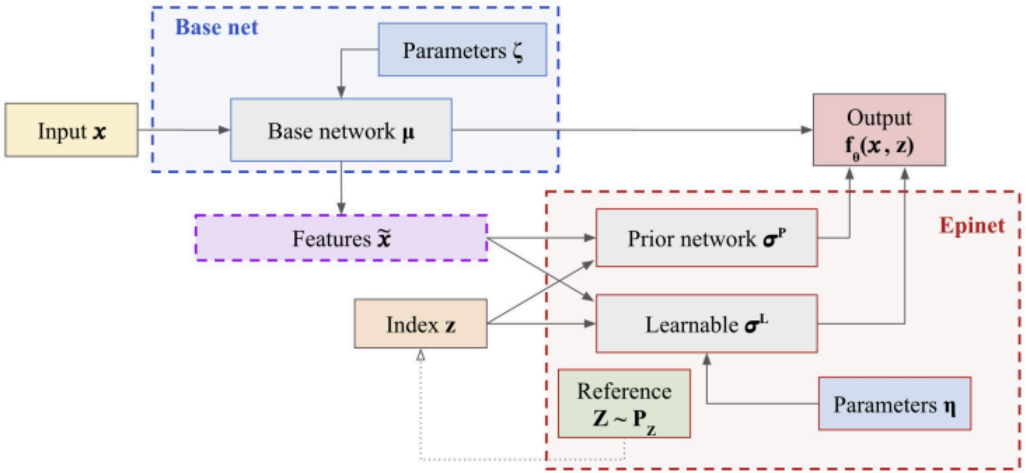


Figure 4: Epinet network architecture.

$$\sigma_\eta(\tilde{x}, z) = \sigma_\eta^L(\tilde{x}, z) + \sigma^P(\tilde{x}, z),$$

- $\sigma_\eta(\tilde{x}, z)$: epinet
- $\sigma_\eta^L(\tilde{x}, z)$: learnable part, 3-layer MLP
- $\sigma^P(\tilde{x}, z)$,: prior net, untrained, simple MLP

Final prediction of Epistemic NN:

$$f_\theta(x, z) = h_\theta(x) + \sigma_\eta(\tilde{x}, z).$$

1. Single Forward Process:

$$f_\theta(x, z)$$

2. Joint Prediction:

$$\hat{P}^{\mathrm{ENN}}(y_{1:n}) = \int_z P_z(z) \prod_{i=1}^{n} \mathrm{softmax}\ (f_\theta(x_i, z))$$

2.4 DoLA + Epistemic NN

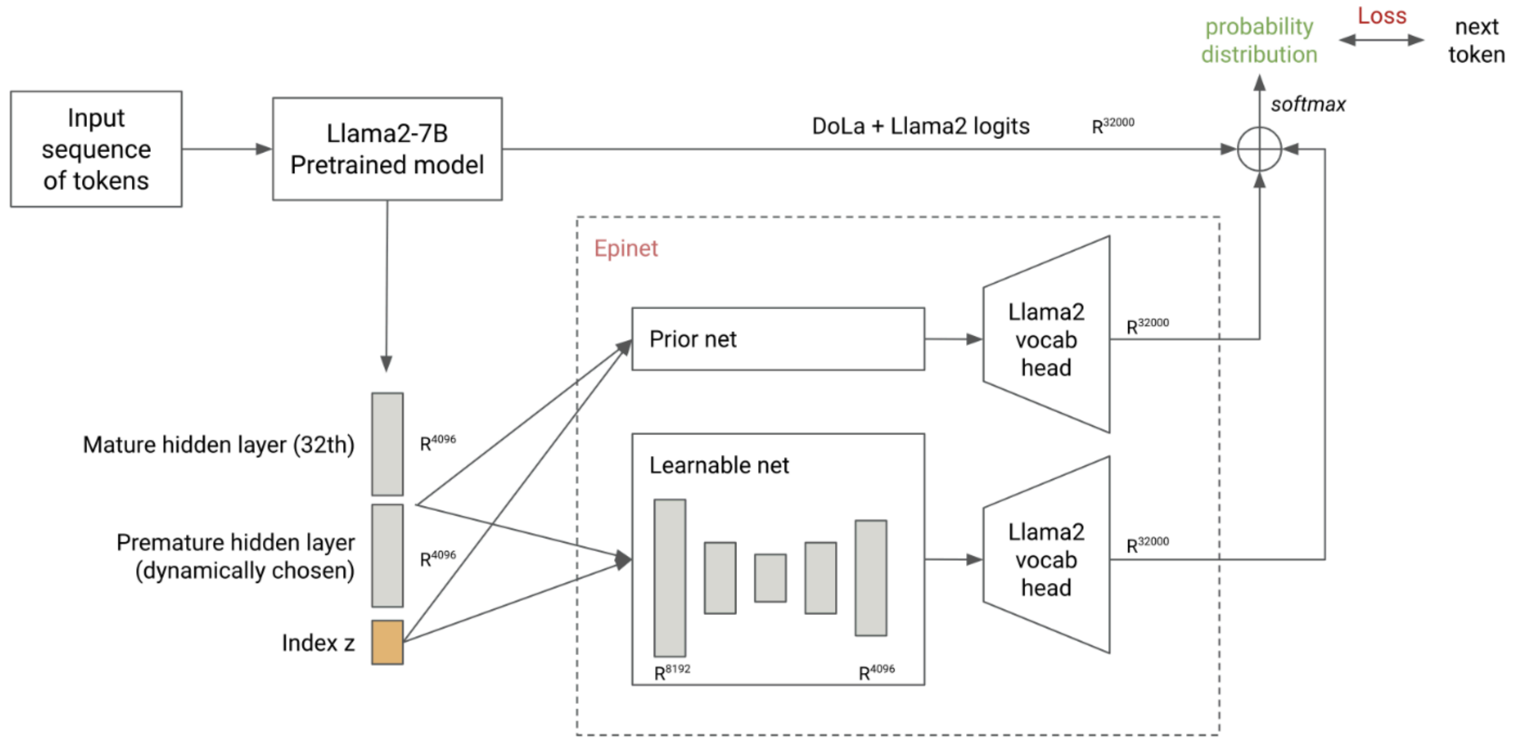
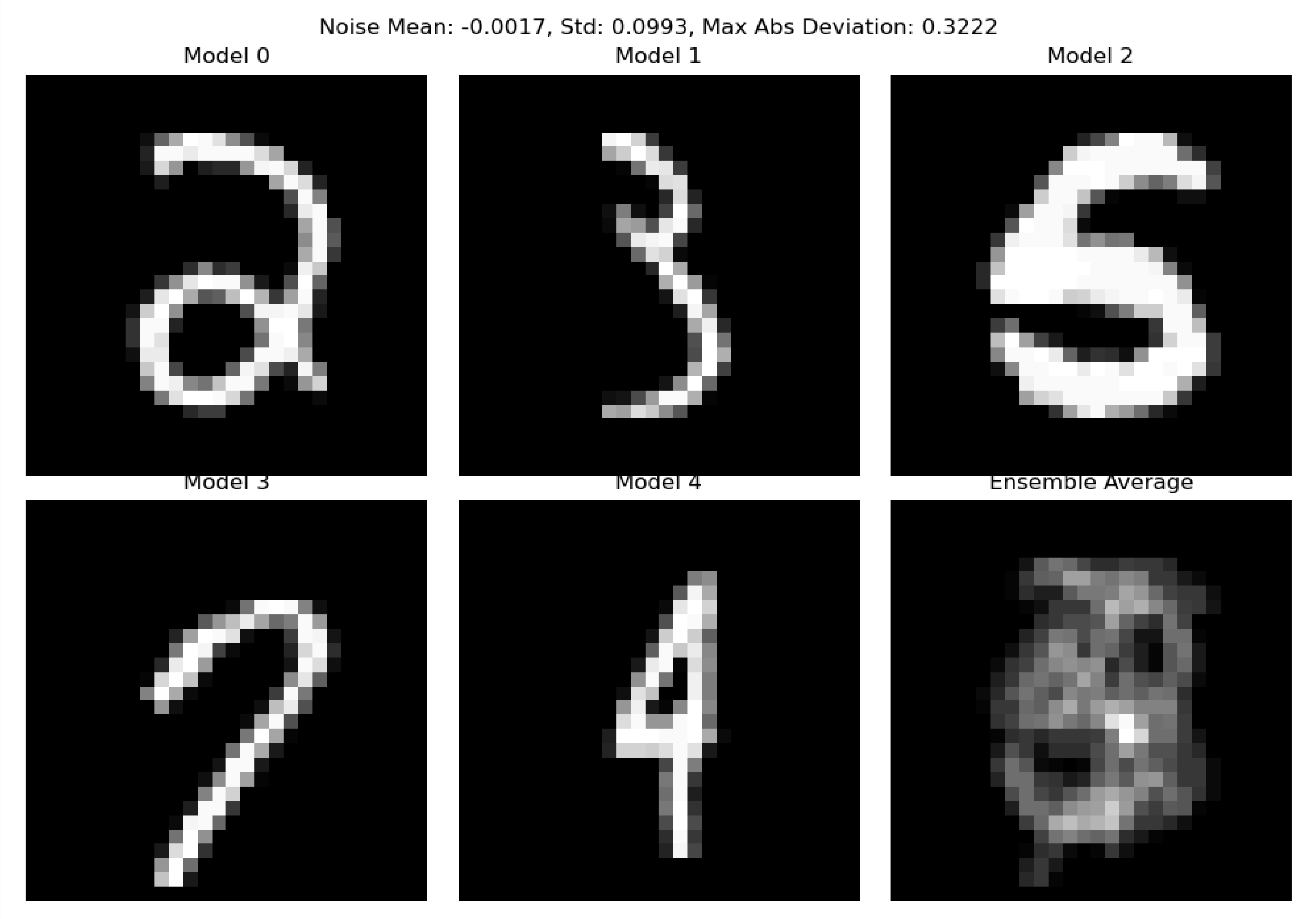


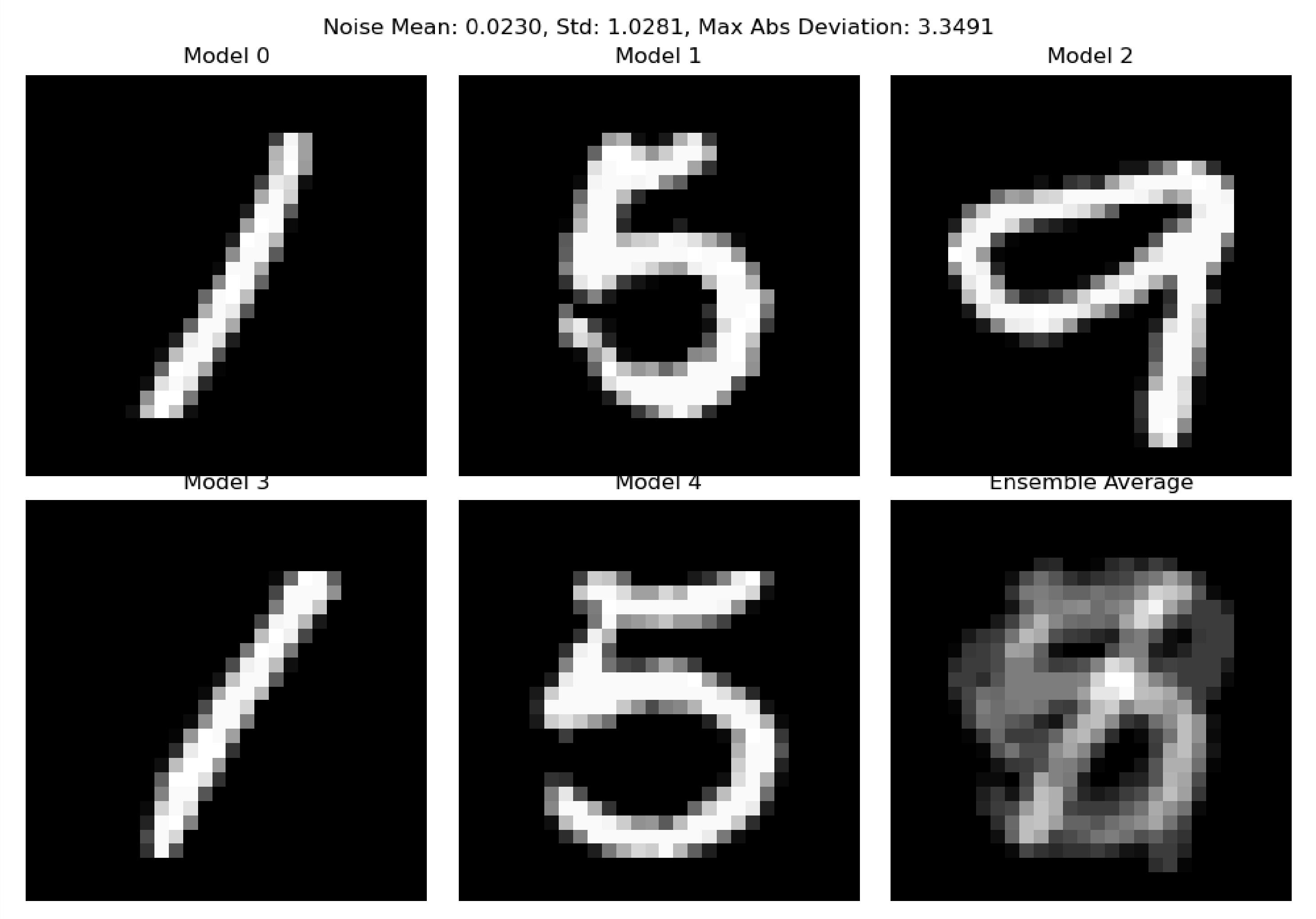
Figure 3: Architecture of the model

$$\text{Final Output} = \text{DoLa Logits} + \sigma^P(\tilde{x}, z)$$

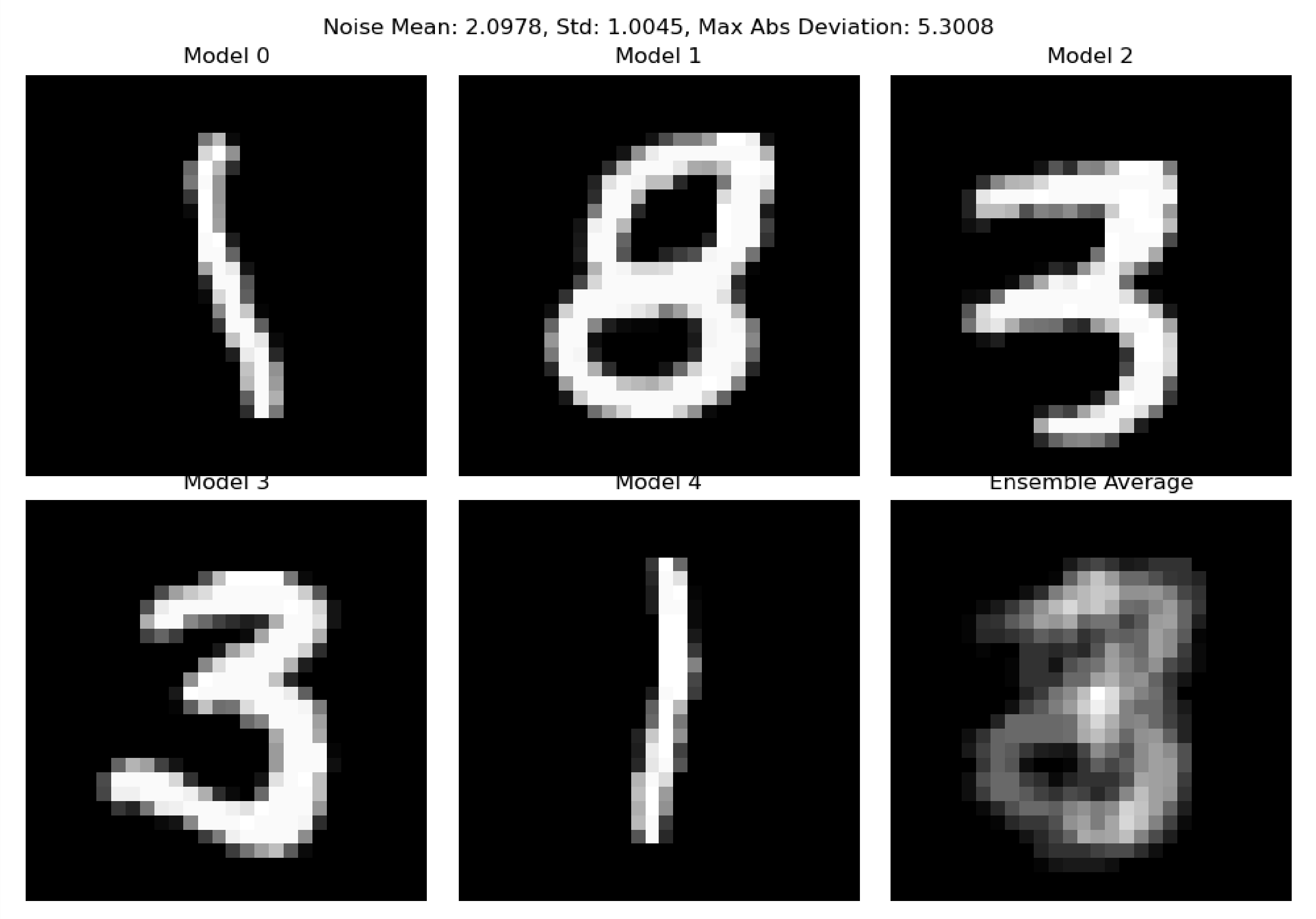
3. About experiments on "Ensemble Diffusion"


Noise Mean: -0.0017, Std: 0.0993, Max Abs Deviation: 0.3222
Model 0
Model 1
Model 2
Model 3
Model 4
Ensemble Average

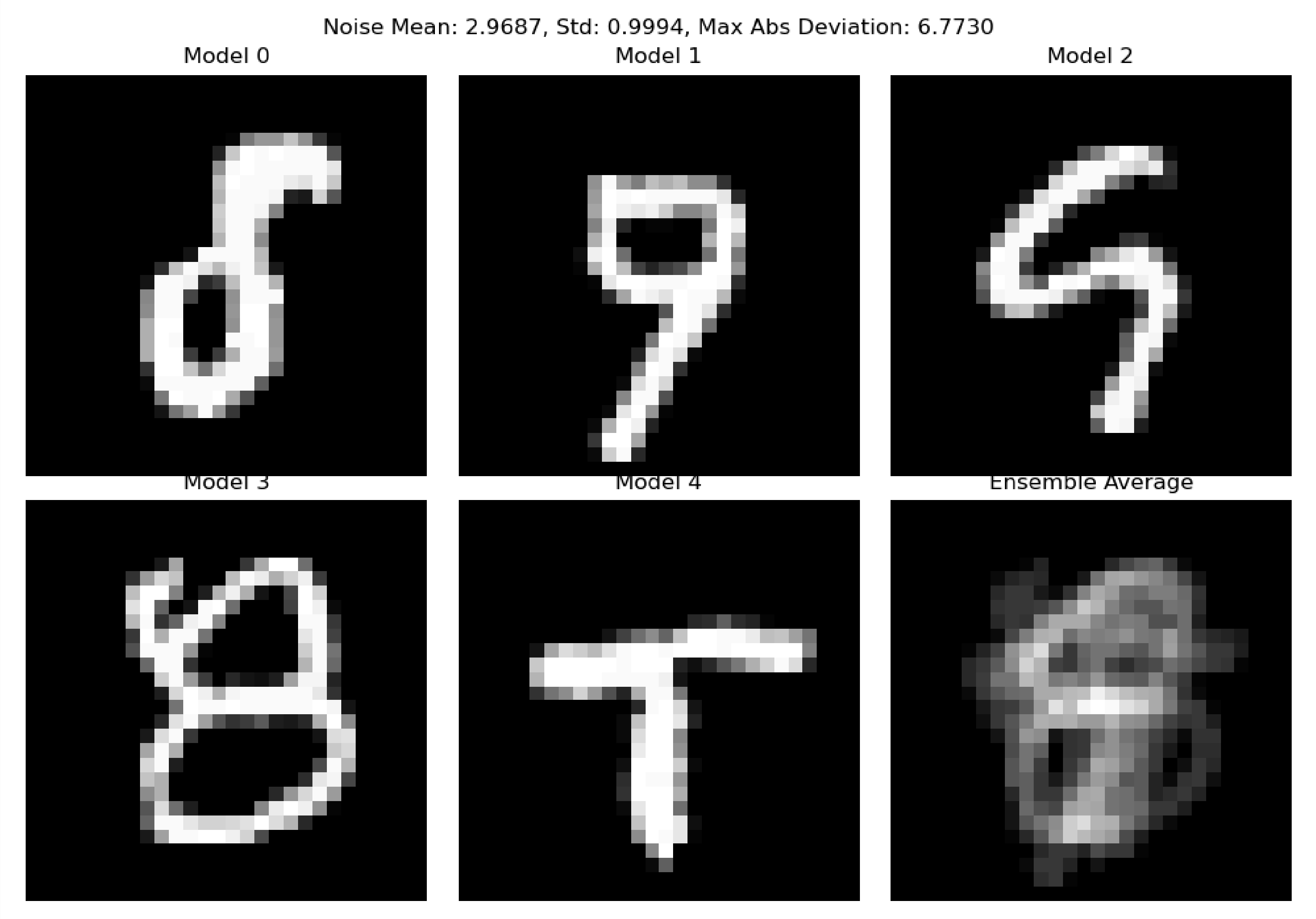
3.1 DDPM results



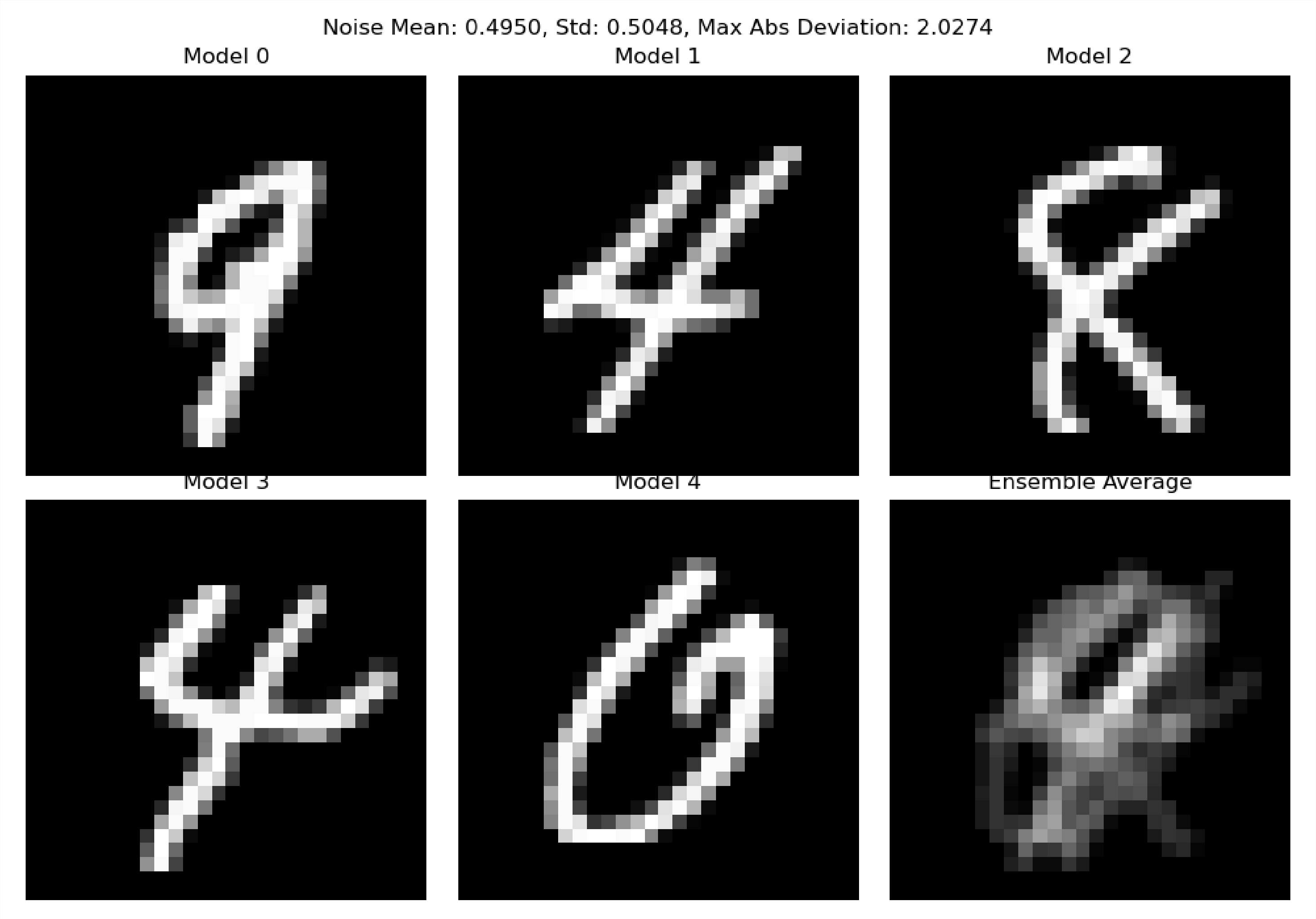
3.1 DDPM results



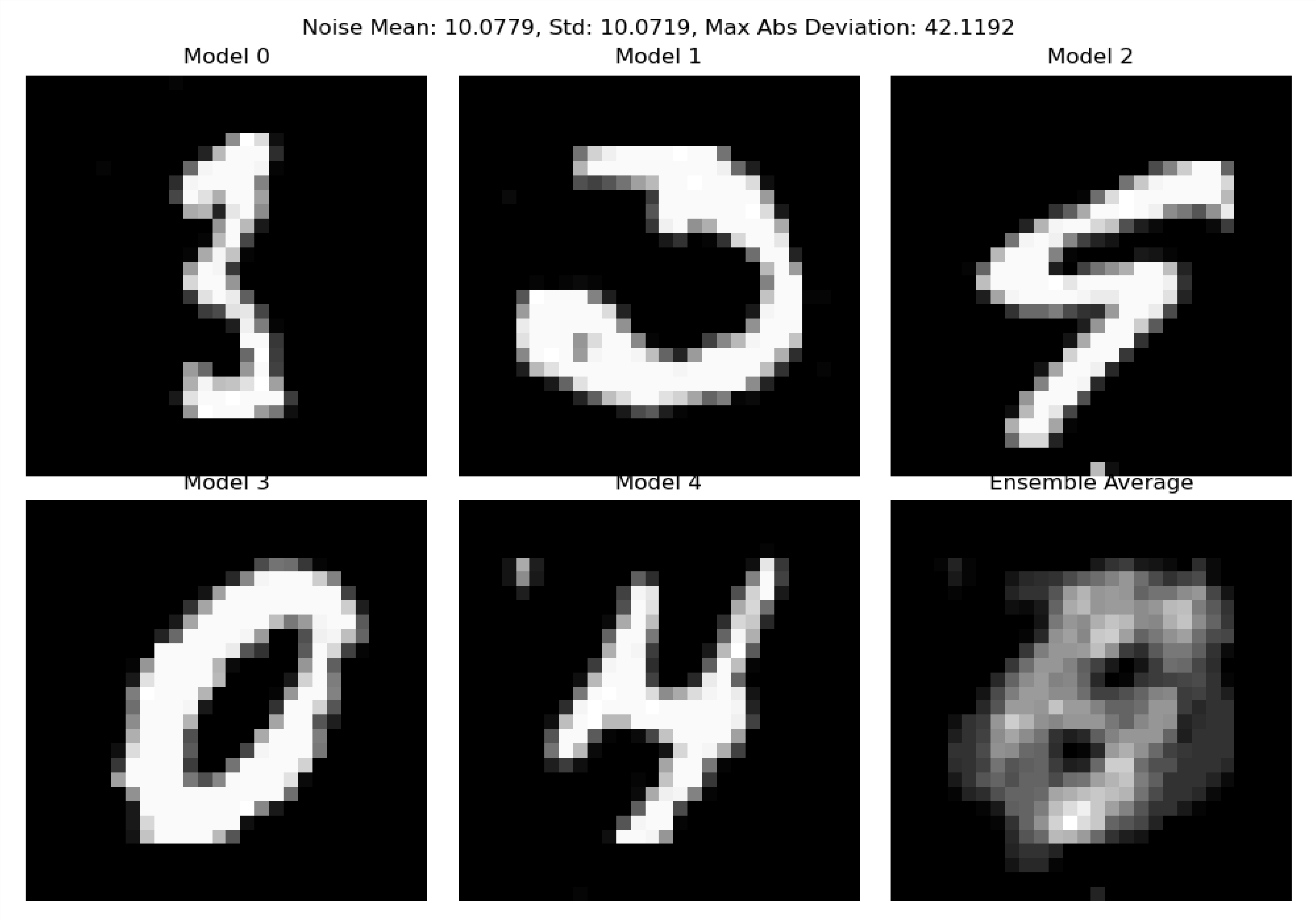
3.1 DDPM results



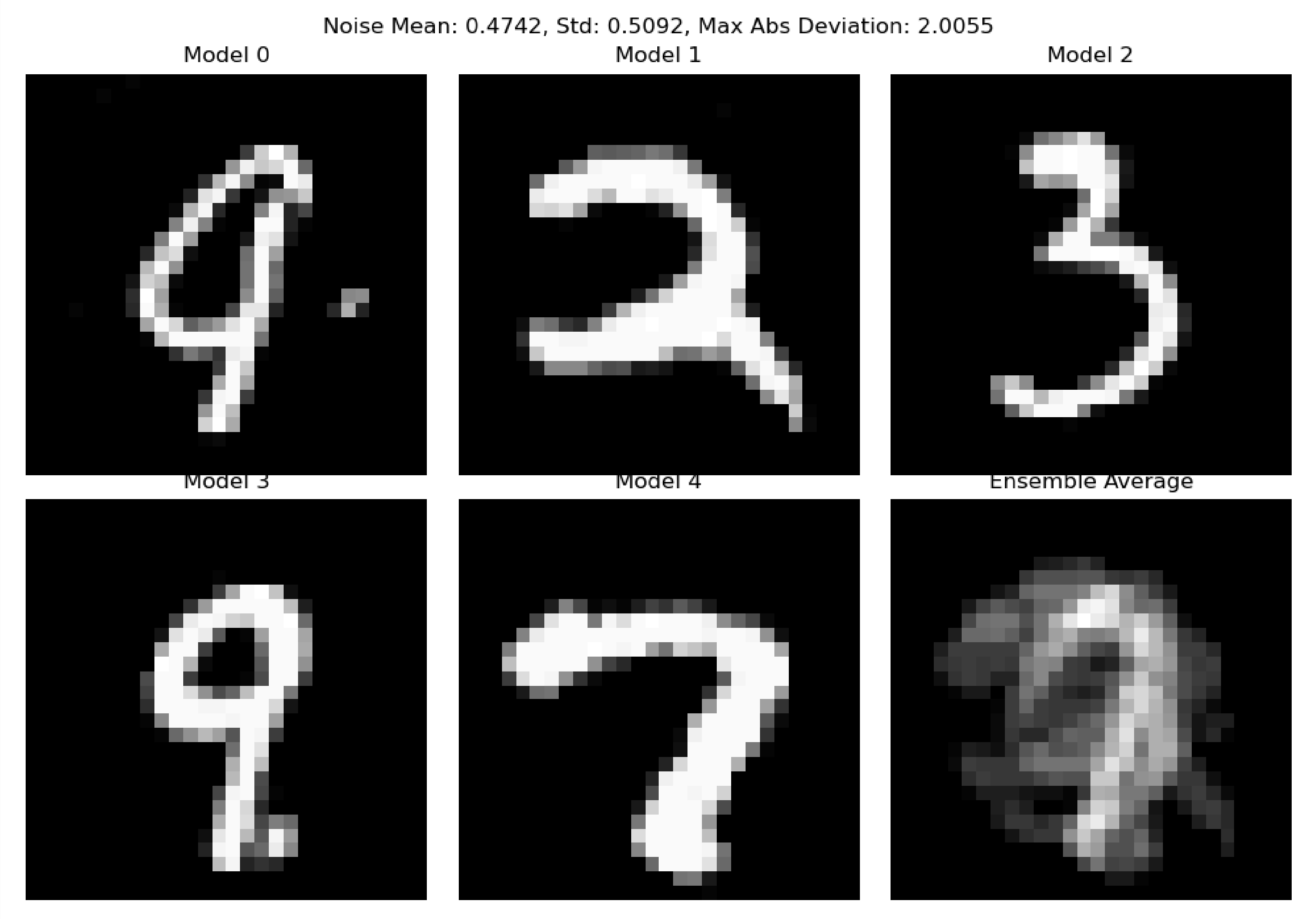
3.1 DDPM results



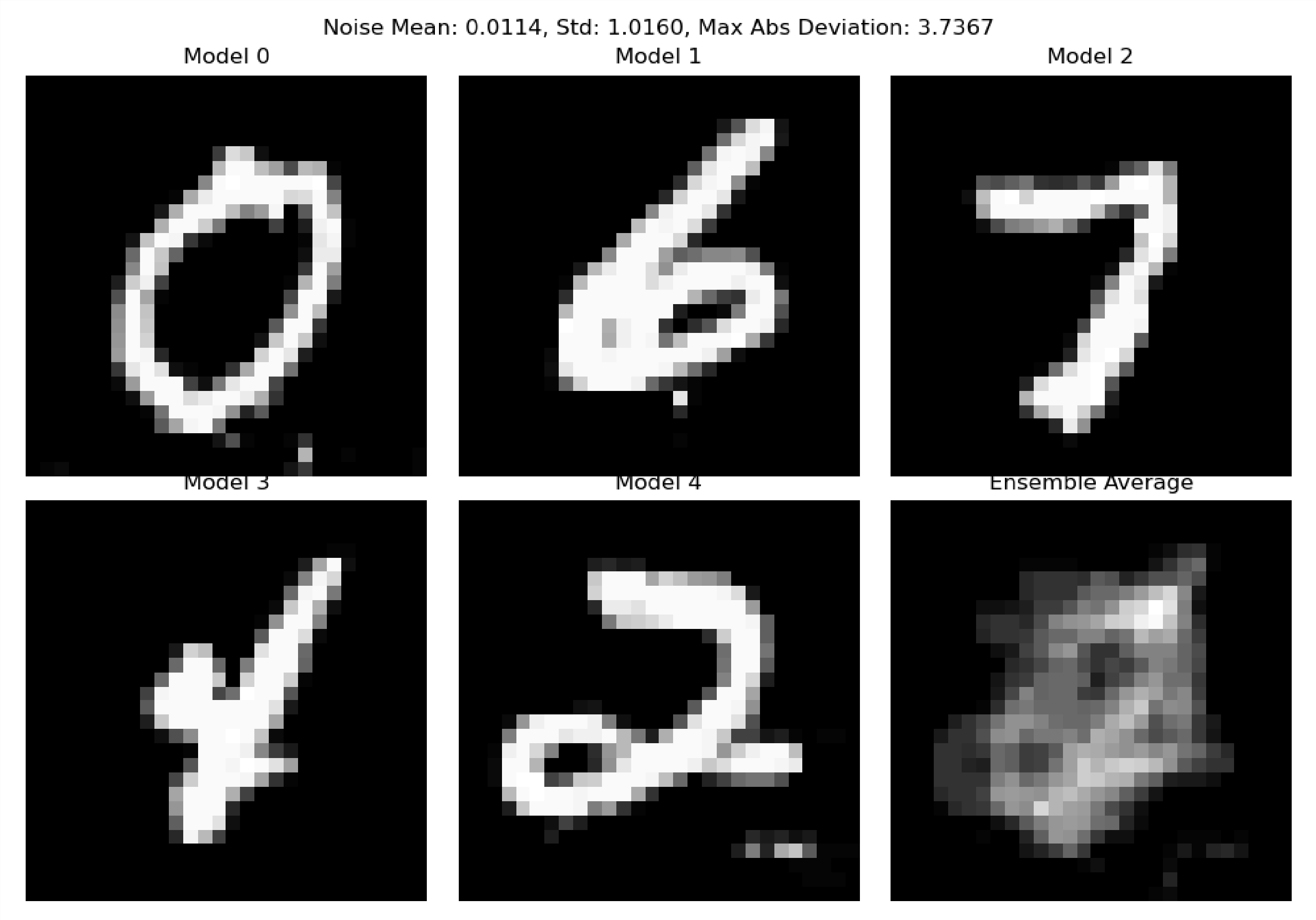
3.1 DDPM results

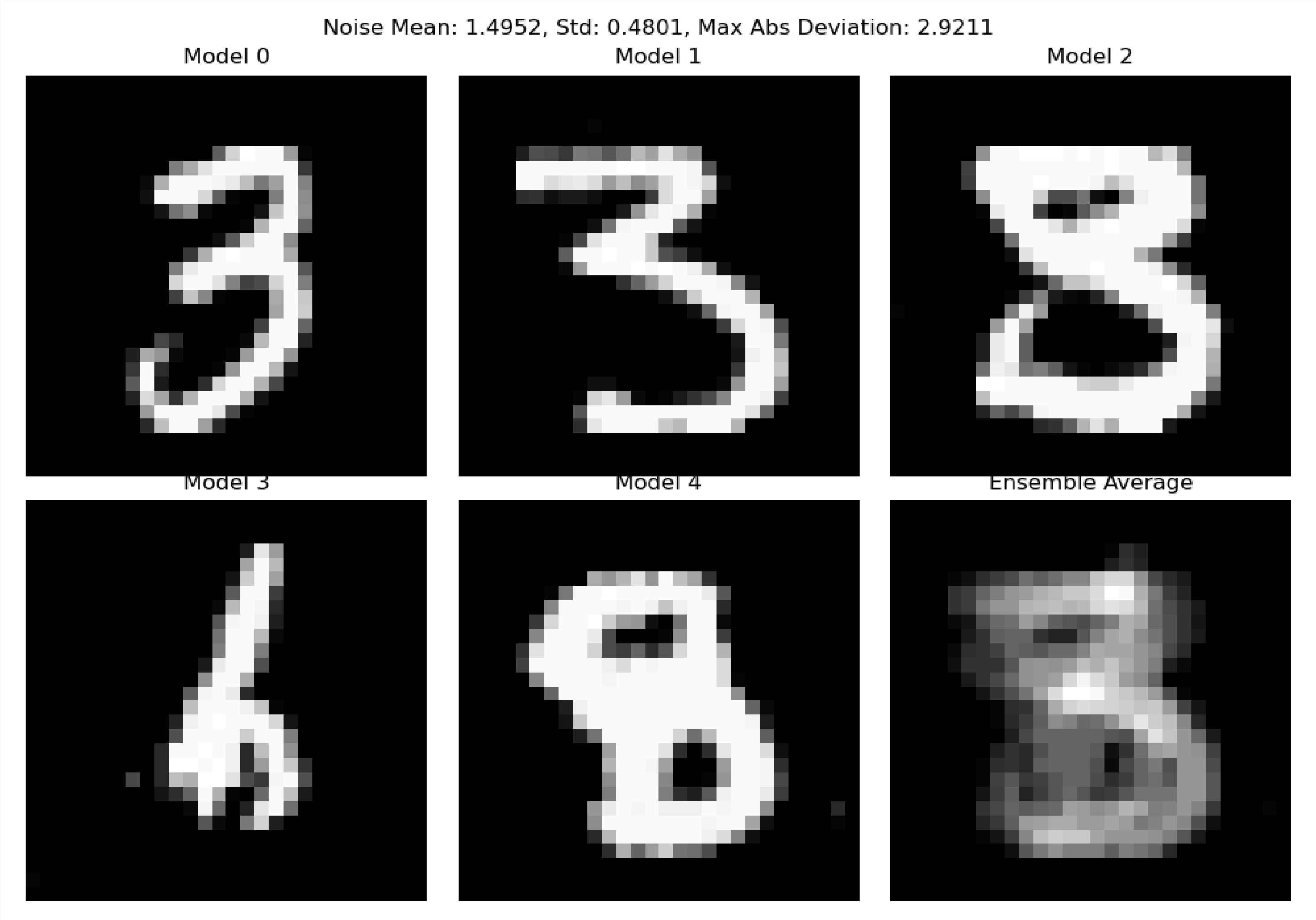


3.2 DDIM results

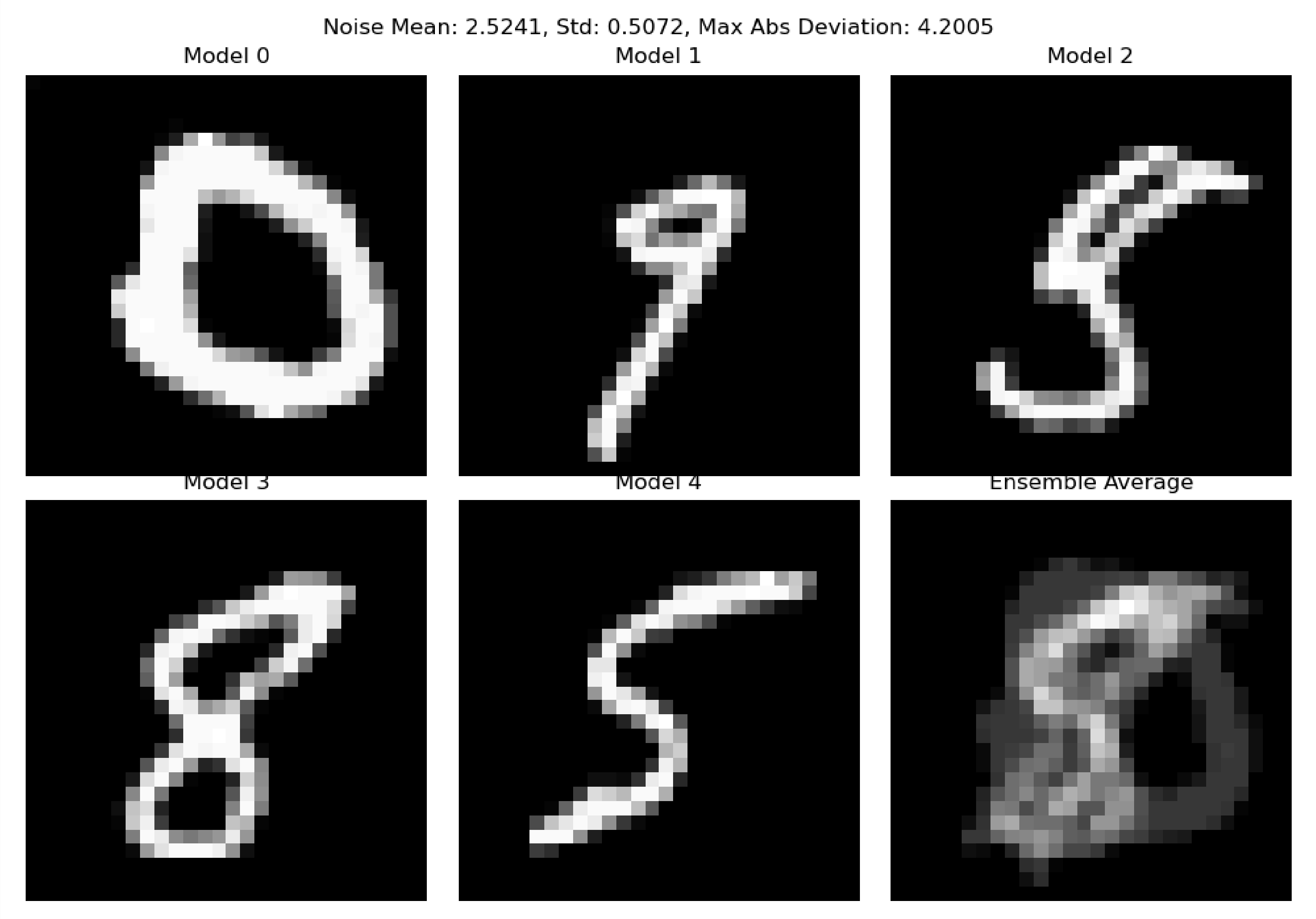


3.2 DDIM results

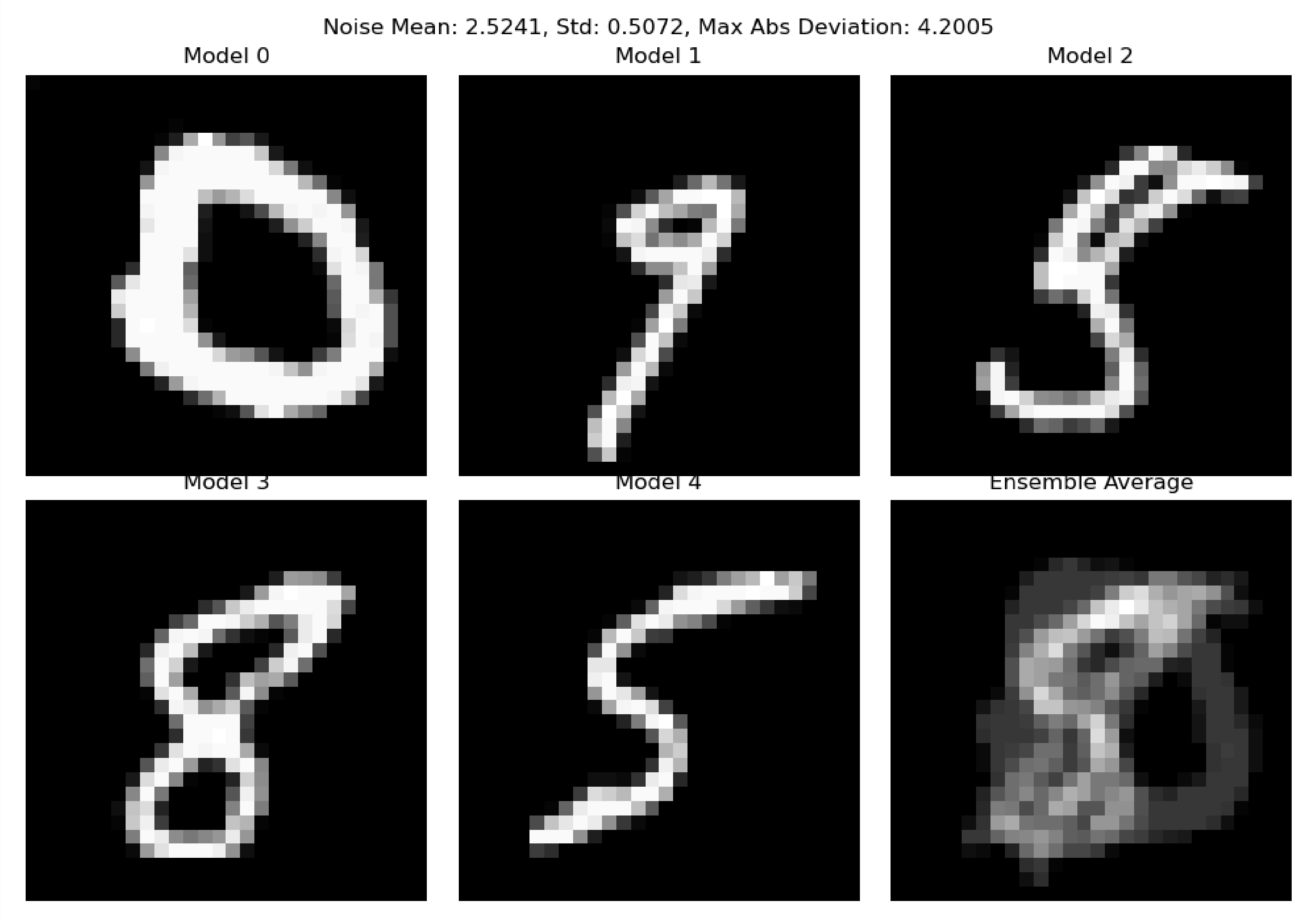


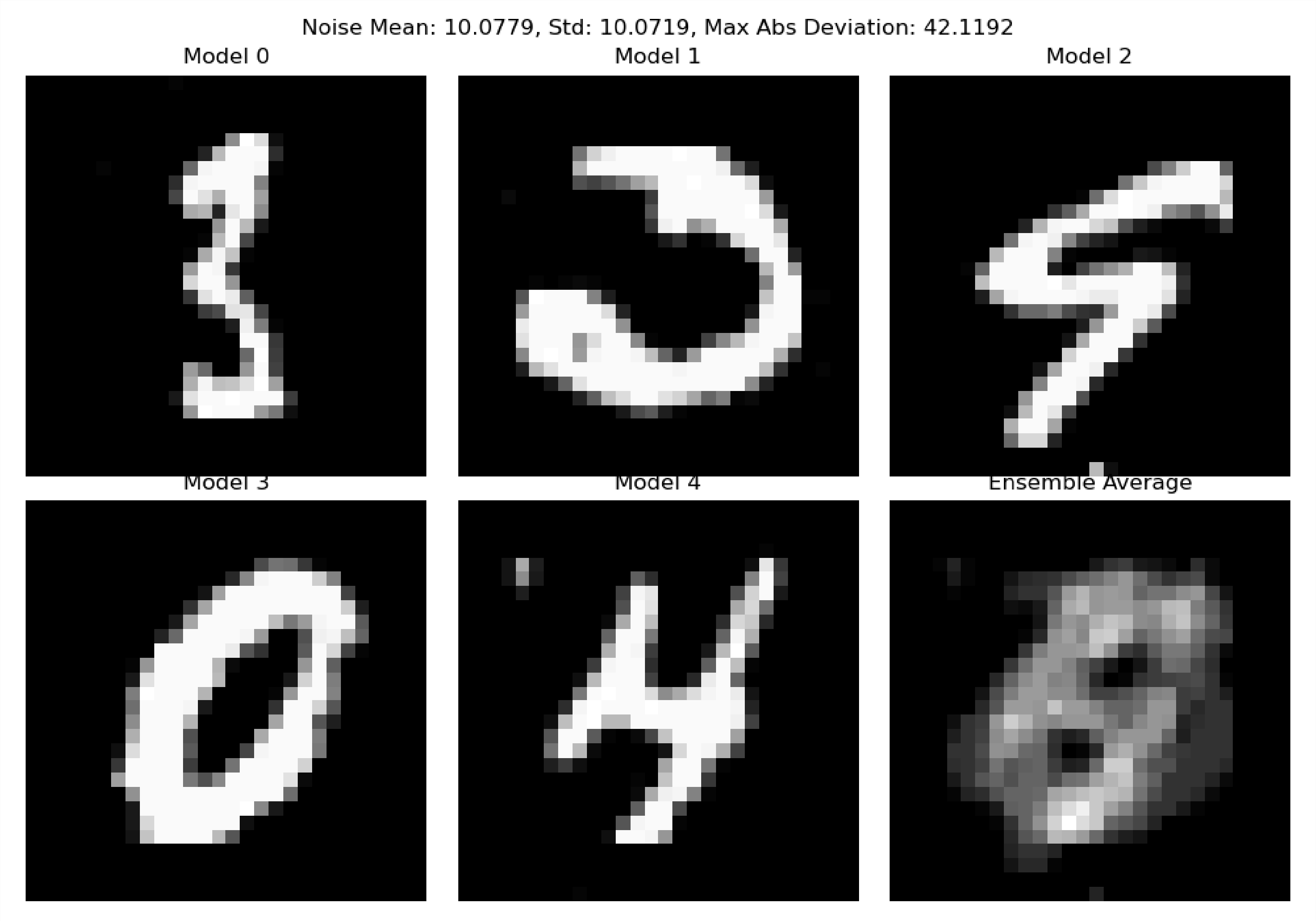

Noise Mean: 1.4952, Std: 0.4801, Max Abs Deviation: 2.9211
Model 0
Model 1
Model 2
Model 3
Model 4
Ensemble Average

3.2 DDIM results



3.2 DDIM results




Noise Mean: 10.0779, Std: 10.0719, Max Abs Deviation: 42.1192
Model 0
Model 1
Model 2
Model 3
Model 4
Ensemble Average

3.3 Next Steps

1. Use 1 model to deonise from T to a mid step, then pass the intermediate image to different denoisers
2. Find some metrics to measure whether the Generated images align with training data (because it is hard to identify visually)