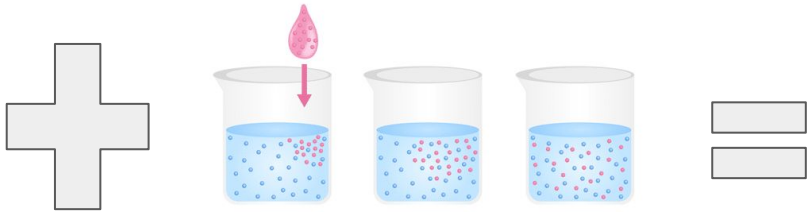
Fine Tuning text diffusion model for russian math tasks

Roman Bokhyan, Timofey Mazurenko

based on the LLaDA 8B model

LLaDA 8B

- consistently outperforms existing diffusion language models by a large margin;
- matches or exceeds top-tier Autoregressive (AR) language models of similar size on the general, math, and coding abilities;
- demonstrates strong planning ability and inference flexibility that naturally benefits from the diffusion modeling.
- regular GPT style transformer architecture



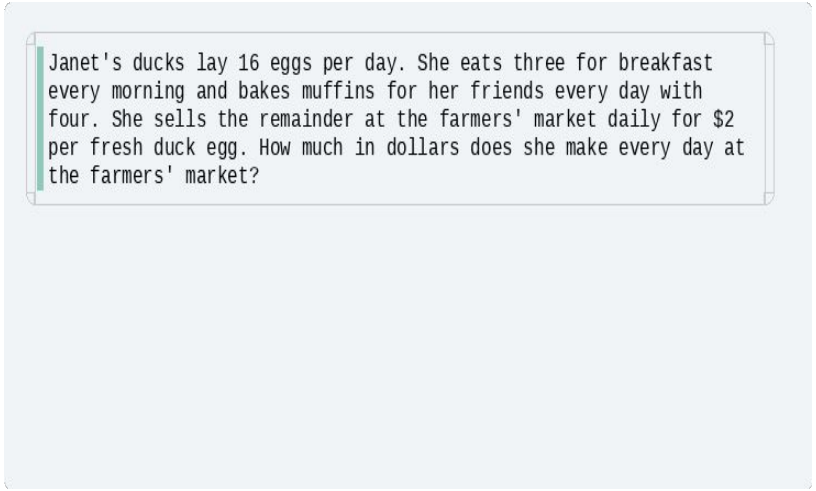
LLaDA 8B: Training

- **The core of LLaDA is a mask predictor, a parametric model pθ(·|x_t) that takes x_t as input and predicts all masked tokens simultaneously. It is trained using a cross-entropy loss computed only on the masked tokens**
- **During training model employs a masking ratio that varies randomly between 0 and 1**

$$\mathcal{L}(\theta) \triangleq -\mathbb{E}_{t,x_0,x_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}[x_t^i=\mathrm{M}]\log p_\theta(x_0^i|x_t)\right],$$

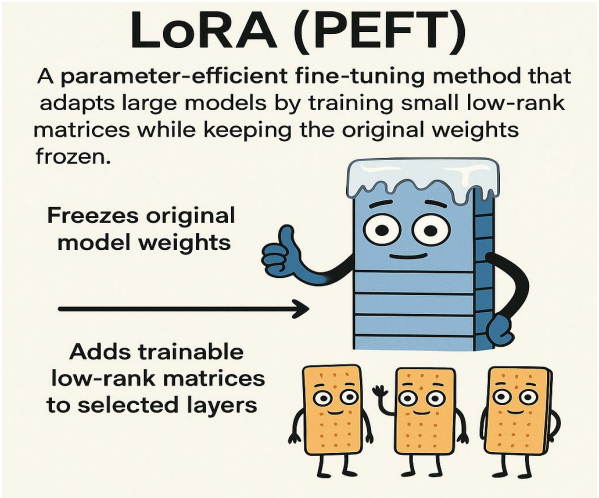
LLaDA 8B: Inference

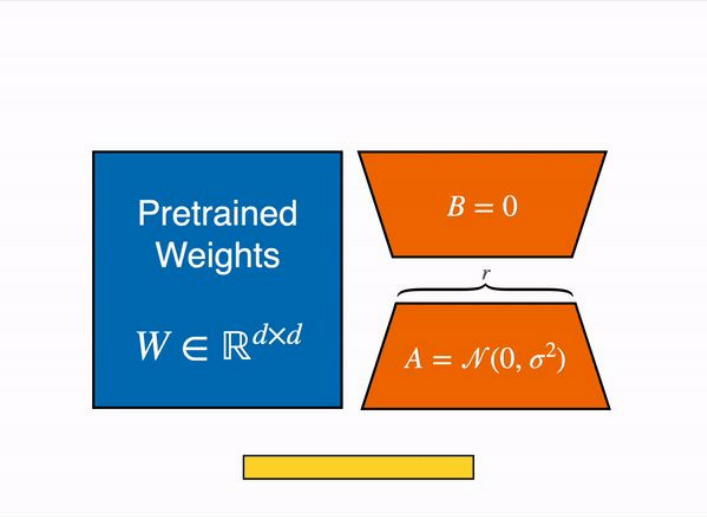
- Once trained, we can simulate a reverse process: choose length of the output n and number of denoising steps k. Create prompt + n |<mask>| tokens. Models unmasks all of them in single pass, then masks back some of the tokens and the process is repeated k times.
- We can mask back tokens randomly, but better way is to take tokens with highest entropy and mask them. That is we mask back tokens in which the model is most unsure.
- Even though we define n, model can generate EOS token in index i << n and then we simply delete tokens starting from i + 1, so model can control output length
- As n increases, both quality of the output and generation time also increase



Parameter-Efficient Fine-Tuning: Low-Rank Adaptation

- Parameter-efficient fine-tuning method for large models
- Freezes original model weights
- Adds trainable low-rank matrices to selected layers
- Trains far fewer parameters than full fine-tuning
- Reduces memory and compute requirements
- Enables multiple task-specific adapters for one base model



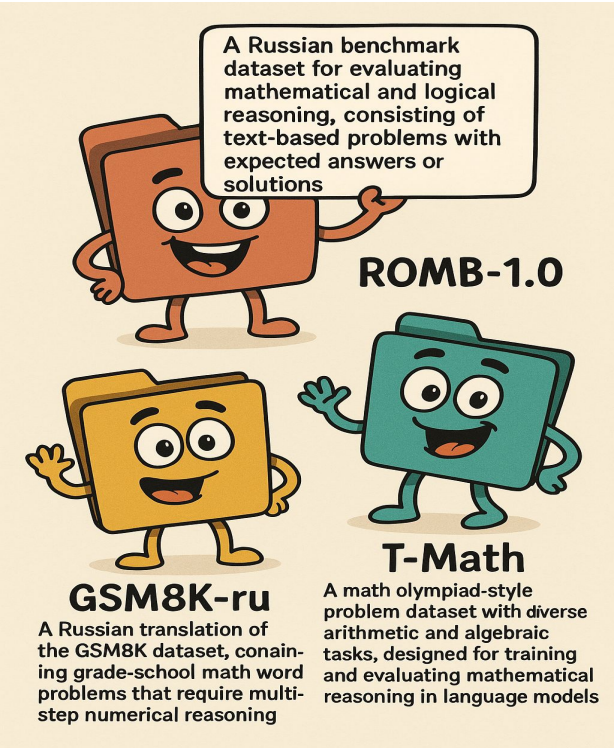


Dataset

ROMB-1.0
A Russian benchmark dataset for evaluating mathematical and logical reasoning, consisting of text-based problems with expected answers or solutions

GSM8K-ru
A Russian translation of the GSM8K dataset, containing grade-school math word problems that require multi-step numerical reasoning

T-Math
A math olympiad-style problem dataset with diverse arithmetic and algebraic tasks, designed for training and evaluating mathematical reasoning in language models

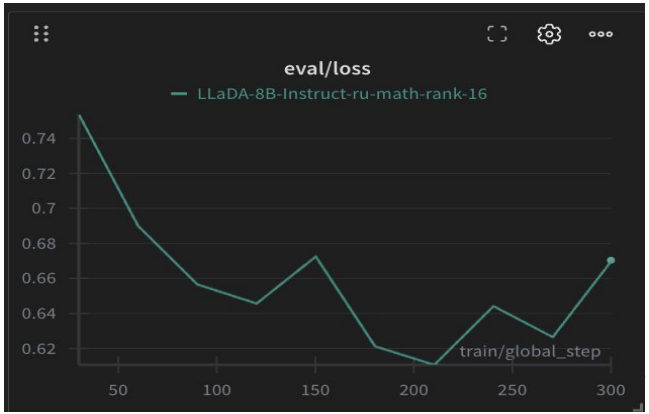


Final split:

- 10k train
- 1k eval

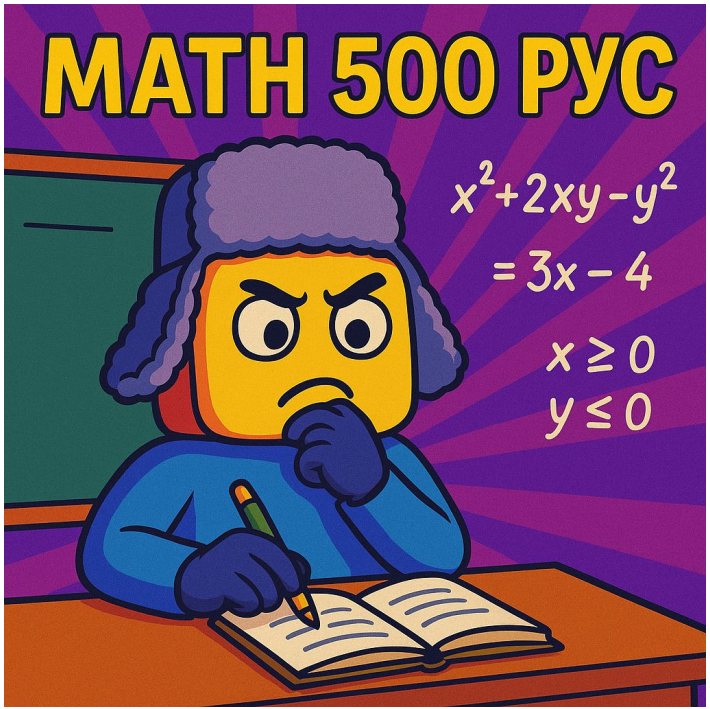
Fine-tuning **LLaDA** to be russian math enjoyer

- rank=16
- target_modules=
  - attention projections
    - q_proj - defines what each token attends to
    - k_proj - defines which tokens are important to attend to
    - v_proj - defines what information is passed through attention
    - o_proj, – mixes attention heads into the final representation
  - FFN projections
    - up_proj - expands hidden states to a higher-dimensional space
    - gate_proj - controls information flow via gated nonlinearity
    - down_proj - projects FFN output back to the model dimension
- dropout=0.05
- lr=2e-4
- 1 epoch



RU-MATH-500

- **A benchmark of 500 challenging math problems**
- **Focused on multi-step mathematical reasoning**
- **Covers arithmetic, algebra, and word problems**
- **Used to evaluate reasoning capabilities of language models**
- **Russian translation provided by Avito**



Experiments during inference on benchmarks

experiment with different number of denoising steps [150, 100, 50]

fix the following parameters

- 'max_new_tokens': 512
- 'temperature': 0.0,
- 'remasking': 'low_confidence'

- Base - base LLaDA Model (no sft)
- LoRA - LLaDA after sft

run	accuracy	seconds_per_task
Base 150	0.162	7.96
Base 100	0.148	5.31
Base 50	0.098	2.66
LoRA 150	0.162	9.82
LoRA 100	0.126	6.53
LoRA 50	0.090	3.27

Performance comparison with same size models

Model	Qwen3-8B	Avibe	RuadaptQwen3-8B-Hybrid	Base 150	Base 100	Base 50	LoRA 150	LoRA 100	LoRA 50
Result	0,546	0,686	0,690	0.162	0.148	0.090	0.162	0.126	0.090

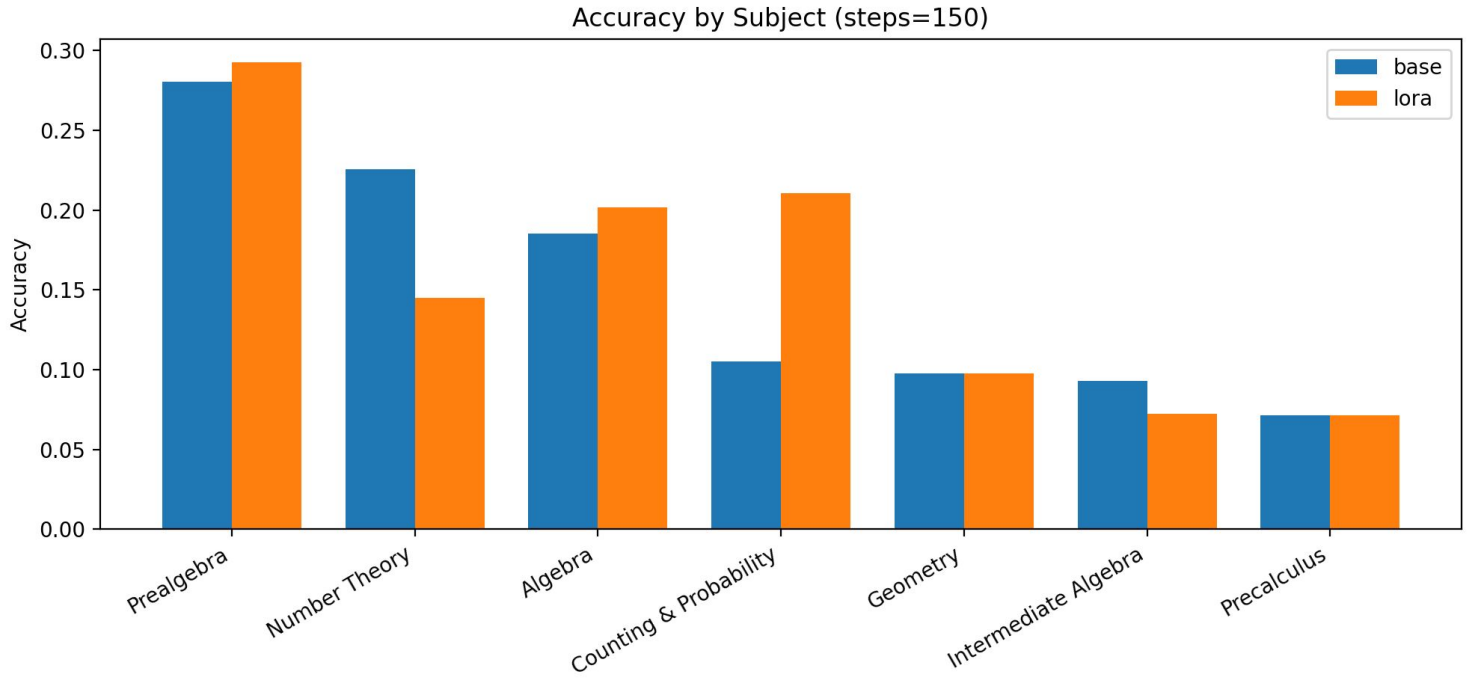
On english MATH500 original model scores:

- Seq len 128: 26.0%
- Seq len 256: 32.4%
- Seq len 512: 36.2%

Model lacks overall russian understanding, because it performs ~2 times better in english. To increase the results substantially, more russian text would be needed (and probably not only math)

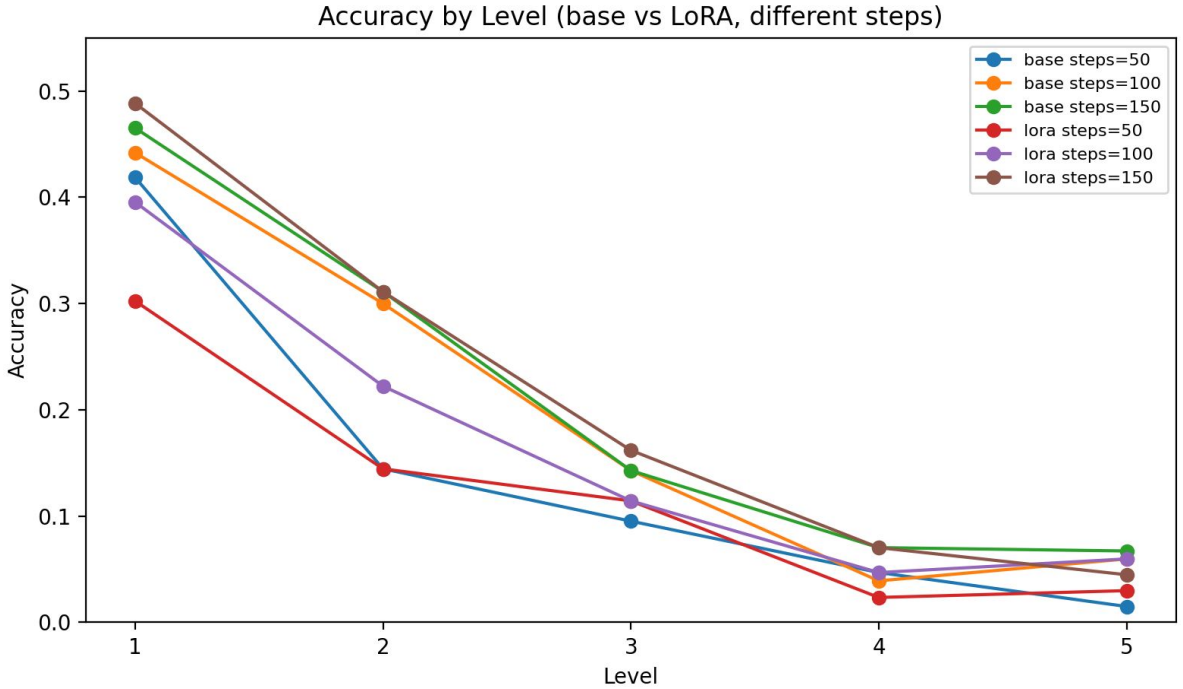
Per subject analysis

Our model wins on some domains, that is likely that those domains were heavily represented in the dataset (like probability for example), our SFT helped to substantially increase the quality. Hence we hypothesize, that if we had more qualitative data for another subjects this trend would generalize



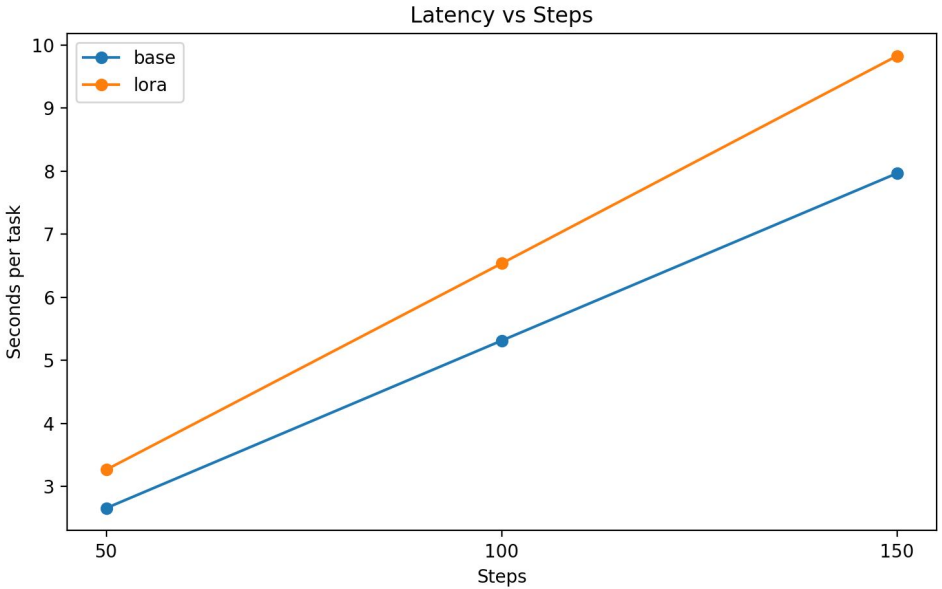
Difficulty of tasks (level) analysis

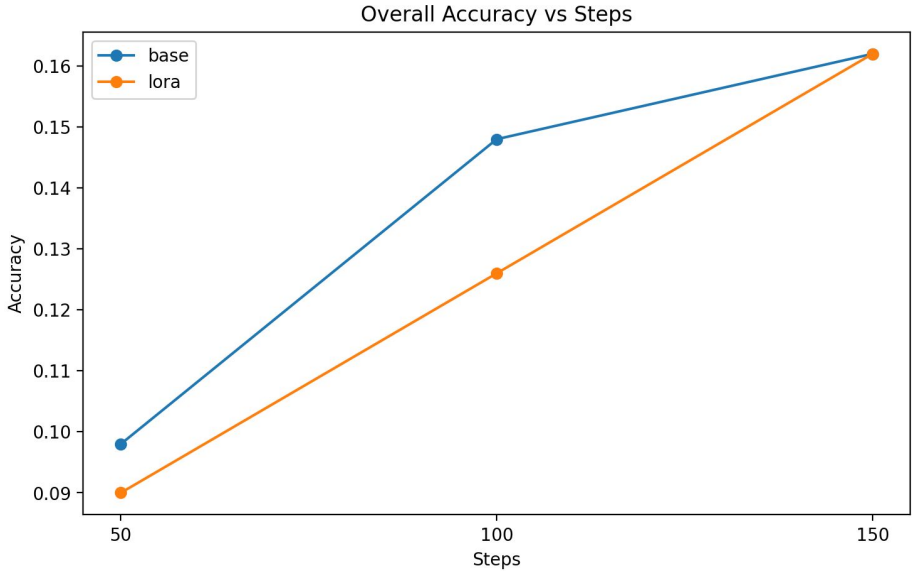
Our model is also clearly better as easier tasks (level 1, 2, 3), and not worth on harder tasks (level 4, 5)



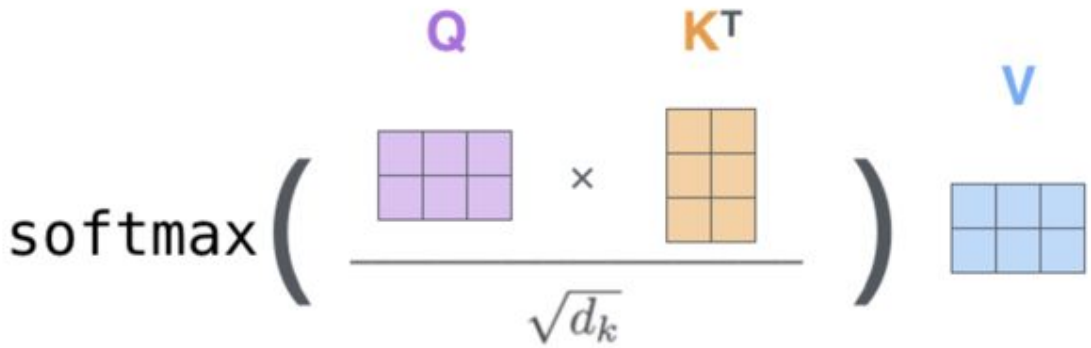
Performance analysis

Latency increases linearly with steps. More steps results in higher accuracy





Thanks for your



data, model and scripts to reproduce our results can be found here: github