Sampler-Conditioned Diffusion Language Models

Madison Davis
Harvard College
madisondavis@college

Kento Nishi
Harvard College
kentonishi@college

Harrison Zhang
Harvard College
hzhang1@college

Victor Cai
Harvard College
victorcai@college

***Abstract*—Large language model specialization typically involves routing, mixture-of-experts (MoE), or separate fine-tuned checkpoints, duplicating parameters across domains. Diffusion language models (DLMs) support specialization at sampling time via iterative denoising and classifier-free guidance. We study whether a sampler-conditioned DLM (SC-DLM), equipped with domain embeddings and guidance, can compose multiple capabilities at sampling time and compete with autoregressive (AR) baselines under a fixed parameter budget. We implement six ≈120M parameter models with matched compute per sample and identical data pipelines: Joint ARM, MoE ARM, Separate ARMs (half of the parameter budget per domain), Joint DLM, Separate DLMs, and a Conditional DLM. All are pretrained on FineWeb-Edu-10B and evaluated on GSM8K (math reasoning) and CBT (reading comprehension) datasets. At this small scale, we find that AR models outperform DLMs. However, within the diffusion regime, the Conditional DLM consistently improves over an unguided Joint DLM. Sampling-time domain conditioning is empirically useful inside the diffusion family but does not yet replace AR specialization. The project code can be found here: github.com/Madison-Davis/cs2420-cs2823R-final-project.**

I. Introduction

Problem Monolithic LLMs can underperform on specialized domains. Providers responded by training math, code, or chat variants. These strategies (joint multidomain models, router/MoE, and domain-specific fine-tunes) all duplicate core circuits such as syntax, world knowledge, and basic reasoning, fragmenting capacity and complicating maintenance.

Motivation We seek a mechanism that keeps one intact model and lets capabilities be invoked and modulated without cloning/partitioning weights. DLMs move specialization from parameters to the sampler; generation unfolds through iterative denoising, and classifier-free guidance [4] blends unconditional and conditional predictions at every step, turning a domain embedding into a controllable trajectory influence.

Goal We ask: can a single DLM compose multiple capabilities at sampling time well enough to rival training-time AR specialization, under a fixed parameter budget and matched compute per sample? We use two superficially dissimilar domains that plausibly share structure—EasyMath (GSM-style math) and the Children's Book Test (CBT) for creative/narrative cloze—and compare six ≈120M-parameter models under a common FineWeb-Edu-10B pretraining pipeline.

Leading Figure Figure 1 and Table I show the six models and their validation losses. Architectures differ only in (i) AR vs. DLM and (ii) conditioning strength. AR models occupy the lowest-loss region; the Joint DLM is worst; the Conditional DLM consistently improves over the unguided model but remains behind AR models.

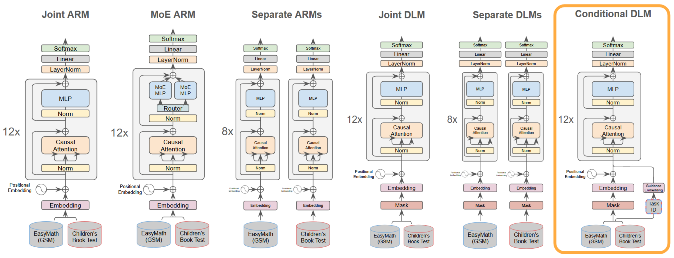

Fig. 1. System overview (Joint ARM, MoE ARM, Separate ARMs, Joint DLM, Separate DLMs, Conditional DLM) with shared FineWeb-Edu pretraining and EasyMath/CBT fine-tuning. Separate variants train one model per domain. All models have ∼120M total parameters and identical I/O.

Novelty Classifier-free guidance [4] was introduced to steer the trajectory of generative image diffusion models without training an auxiliary classifier. Inspired by that mechanism, we inject domain embeddings at sampling time to bias every denoising step toward a math or narrative trajectory while still sharing all weights. For intuition, an unguided DLM trying to solve a math problem partially behaves as if it also needs to write a story (averaging over all domains it has seen); adding the math guidance embedding nudges each denoising step toward math-appropriate reasoning, thereby cutting validation loss. We show in the following sections that this simple sampler conditioning is already helpful.

Contributions

1) We design a Conditional DLM (SC-DLM) that uses domain embeddings for sampling-time specialization without duplicating parameters [II].
2) We implement a six-model comparison at matched parameter budget and compute, isolating architecture and conditioning [III].
3) We show that sampler conditioning improves diffusion (the Conditional DLM is better than the Joint DLM) but that ARMs remain superior overall at this scale [IV].

II. Approach

Description: Model Family All models share:

1) total parameter budget ≈ 120M,
2) matched compute/token (depth, width, sequence length),
3) identical tokenizer, dataloaders, and train/val splits,

4) the same pretraining and fine-tuning schedule.

We study the following models:

1) **Joint ARM (Model 1):** one GPT-style autoregressive model trained on EasyMath and CBT jointly, without an explicit domain token.
2) **MoE ARM (Model 2):** same trunk; MLPs replaced by MoE blocks and a router.
3) **Separate ARMs (Model 3):** two separate ∼60M autoregressive models, each fine-tuned on one domain.
4) **Joint DLM (Model 4):** one diffusion LM trained jointly on both domains; no guidance embedding at sampling.
5) **Separate DLMs (Model 5):** two ∼60M diffusion models, each trained on a single domain without guidance.
6) **Conditional DLM (Model 6):** same diffusion backbone; domain embeddings for sampling-time conditioning.

Models differ only in architecture (ARM vs. DLM) and how domain identity enters the model. For ARMs, conditioning strength increases from implicit multidomain (1) to MoE (2) to separate domain-specific fine-tunes (3). For DLMs, we compare shared multidomain training (4), per-domain specialists (5), and sampler conditioning (6).

Conditional DLM Denoiser. Our sampler-conditioned model follows the discrete Block Diffusion process [5]. Tokens are grouped into blocks of size $B = 16$. For each block b we sample a corruption rate $t_b \in [t_\ell, t_u]$ and then draw per-token Bernoulli masks with probability t_b, so that every position in the block shares a noise rate but is masked independently. The corrupted prefix is

$$\mathbf{x}_t = \mathbf{m}_t \odot \mathbf{x}_{\text{mask}} + (1 - \mathbf{m}_t) \odot \mathbf{x}_0, \tag{1}$$

where $\mathbf{m}_t \in \{0, 1\}^T$ stores these Bernoulli draws. We concatenate $[\mathbf{x}_t; \mathbf{x}_0]$ and apply the Block Diffusion attention mask (block-diagonal, offset block causal, block causal) so a noisy position can read its own block, already-denoised blocks in the same document, and the clean suffix. The transformer outputs logits $\ell_\theta(\mathbf{x}_t, t, \mathbf{p}, \mathbf{e}_c)$ on masked tokens, and we optimize the block-weighted cross-entropy objective

$$\mathcal{L}_{\text{block}} = \mathbb{E}\left[\frac{\mathbf{m}_t}{t_b + 10^{-3}} \cdot \mathrm{CE}\big(\ell_\theta(\mathbf{x}_t, t, \mathbf{p}, \mathbf{e}_c), \mathbf{x}_0\big)\right], \tag{2}$$

which emphasizes heavily corrupted blocks. $\mathbf{e}_c$ is a learned domain embedding (math vs. CBT) added to the token embeddings whenever we train or evaluate in guidance mode; runs without conditioning set $\mathbf{e}_c = \mathbf{0}$.

At sampling, we choose the desired domain token (math or CBT) and run the denoiser once per block prefix; guided and unguided generations differ only in whether $\mathbf{e}_c$ is set to the learned embedding or to zero. This keeps specialization entirely within the guidance embedding while maintaining a single shared backbone.

Intellectual Points and Novelty By specializing during sampling time, the Conditional DLM (SC-DLM) reuses the same weights for all domains, preventing fragmentation of parameters into separate models/experts and allowing the model to learn shared circuits only once between domains for parameter efficiency and cross-domain transfer. SC-DLM can effectively reuse the same learned weights in different domain contexts.

III. Implementation

Planned Experiments We study the pairwise comparisons below within each architecture family and qualitatively assess sample efficiency (optimization steps to reach a given loss) and parameter efficiency (loss at fixed ∼120M budget):

1) **Model 2 vs 1:** Does adding router/MoE conditioning to the Joint ARM improve performance?
2) **Model 3 vs 1:** Do Separate ARMs outperform the shared Joint ARM?
3) **Model 3 vs 2:** Are explicit domain splits better than router-based specialization for ARMs?
4) **Model 5 vs 4:** Do Separate DLMs beat the Joint DLM?
5) **Model 6 vs 4:** Does conditioning help diffusion under a fixed 120M budget?
6) **Model 6 vs 5:** Are guidance embeddings more effective than duplicated diffusion checkpoints?

Models Models 1–2 are GPT-2–style Joint/MoE ARMs with 12 layers, 768-d embeddings, 12 heads, 1024 context, and ∼124M parameters (Model 2 replaces MLPs with MoE blocks and a router). Model 3 (Separate ARMs) uses two 8-layer, 512-d, 8-head autoregressive models (∼60M each, ∼120M total). Model 4 is the Joint DLM with a 12-layer Transformer denoiser and 640-d embeddings (∼123M parameters). Model 5 (Separate DLMs) mirrors Model 3 with two 8-layer, 512-d diffusion models trained independently per domain. Model 6 (Conditional DLM) reuses the Joint DLM backbone but adds a two-vector domain-embedding table (∼1.3k parameters) for sampler conditioning.

Datasets All models are pretrained on FineWeb-Edu-10B, then fine-tuned on:

1) **EasyMath**: GSM word problems; we score answer tokens.
2) **CBT**: narrative cloze; we predict masked tokens in passages.

Training, Systems, and Tools We use AdamW with cosine learning-rate schedules. Pretraining runs ∼1000 iterations on FineWeb; fine-tuning adds ∼2000 iterations on domain data, with lower LR and higher dropout for larger models. Runs use Harvard FASRC and MIT Engaging clusters; configs are shared across models. DLM objectives are evaluated directly on masked-token denoising (no DDIM sampler); sampler conditioning is toggled by selecting the math or CBT guidance embedding during finetuning and evaluation. Model definition, training loops, and dataloaders are developed in Python/PyTorch. We set up single-node DDP pre-training and fine-tuning with torchrun on multiple L40S GPUs. We use training clusters from FASRC and MIT Engaging.

IV. Results

Table I summarizes our main quantitative results.

Model	Arch	Scale	Combined	CBT	GSM
1	AR	124M	1.86	3.09	0.645
2	AR	124M	1.80	2.97	0.646
3	AR	2×60M	1.75	2.84	0.66
4	DLM	123M	6.84	6.70	6.97
5	DLM	2×60M	6.95	6.75	7.15
6	DLM	123M	6.50	6.49	6.50

TABLE I
Validation losses at ∼120M parameters.

AR Conditioning and Architecture Effects Strengthening AR specialization (joint → MoE → separate) yields monotonic gains in validation loss and accuracy, but even the simplest AR model (Model 1) still beats the best DLM: combined validation loss 1.86 (Model 1) vs. 6.84 (Model 4). Architecture remains the dominant factor at this scale.

DLM Domain Specialization (Model 5 vs 4) Training separate diffusion checkpoints yields CBT loss 6.75 and GSM loss 7.15, averaging to 6.95—slightly better on CBT but worse on GSM versus the shared model (6.70/6.97). Duplicating checkpoints does not recover the capacity lost to halving the model size.

DLM Conditioning Effect (Model 6 vs 4/5) The Conditional DLM (Model 6) consistently improves over the Joint DLM (Model 4) on both EasyMath and CBT: combined loss drops from 6.84 to 6.50, CBT from 6.70 to 6.49, and GSM from 6.97 to 6.50. These gains arrive without duplicating checkpoints—guidance embeddings alone provide the needed domain signal, and Model 6 also matches or beats the Separate DLMs (Model 5).

Best-of-Class Comparison Comparing the Conditional DLM (Model 6) to the best ARM setup (Models 2/3), autoregressive models remain clearly superior in absolute performance at ∼120M. The Conditional DLM offers cleaner parameter sharing (one checkpoint, explicit sampler control) but does not reach ARM-level likelihood quality; for any realistic loss threshold, ARMs reach it in fewer optimization steps.

V. Related Work

Diffusion-LM [1] and D3PM [2] showed that text generation can be learned by iterative denoising in continuous or discrete spaces. LLaDA [3] scales discrete DLMs to billions of parameters and reports competitive perplexities with AR LMs, suggesting that diffusion may be parameter-efficient at larger scales. Classifier-free guidance [4] was introduced for image diffusion and is now standard for controlling generation; we adopt its spirit by injecting domain embeddings at sampling time, though we do not mix conditional and unconditional logits. On the AR side, multidomain training, MoE routing, and domain-specific fine-tuning are standard practices; our Models 1–3 mirror these and serve as realistic baselines.

VI. Conclusion

Contributions to Literature We introduced a Conditional DLM (SC-DLM) that performs domain-aware sampling via learned guidance embeddings, and we compared it against realistic ARM baselines and a Joint DLM under a fixed ∼120M parameter budget and matched compute.

Lessons

1) At this small scale, AR models remain substantially better than all diffusion variants on EasyMath and CBT; pairing FineWeb pretraining with cautious finetuning (lower LR, higher dropout) remains essential.
2) Sampler conditioning is empirically useful within diffusion: the Conditional DLM consistently improves over the Joint DLM without duplicating checkpoints.
3) Splitting the diffusion budget into two 60M specialists marginally helps CBT but hurts GSM, so duplicated domain checkpoints do not reclaim the lost capacity.

Self-Assesssment See Planned Experiments [III]. We find that stronger sampler conditioning for DLM and stronger specialization for AR improve validation loss. We could not determine the model size at which DLMs would outperform AR models with the small scale at our disposal.

Future Work With more time and compute, we would: (1) perform size ablations to search for an AR–DLM crossover point; (2) explore learned guidance schedules and blended domain embeddings; (3) extend to richer domain suites (e.g., code, dialogue) to further test sampling-time composition.

Group Contribution Statement

Madison Davis: datasets/dataloaders (FineWeb, EasyMath, CBT), FASRC setup, AR Models 1–2 implementation and training, AR evaluation.
Harrison Zhang: AR Model 3 implementation, single-node DDP training, domain-specific AR evaluation, hyperparameter tuning.
Kento Nishi: diffusion backbone and Conditional DLM implementation, DLM/Conditional DLM training and evaluation, overall experimental design and main write-up.
Victor Cai: model diagrams, visualization, and parameter accounting, experiment tracking and plotting, qualitative analysis, novelty and intellectual points.

References

[1] X. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. Hashimoto, “ Diffusion-LM Improves Controllable Text Generation,” arXiv, 2022.

[2] J. Austin, D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, “Structured Denoising Diffusion Models in Discrete State-Spaces,” Advances in Neural Information Processing Systems, 2021.

[3] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J. Wen, and C. Li, “Large Language Diffusion Models,” arXiv:2407.04907, 2024.

[4] J. Ho and T. Salimans, “Classifier-Free Diffusion Guidance,” arXiv, 2022.

[5] M. Arriola, Z. Qi, A. K. Gokaslan, J. Han, J. T. Chiu, S. S. Sahoo, Z. Yang, and V. Kuleshov, “Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models,” arXiv:2503.09573, 2025.