Compositional Diffusion Language Models: Scaling Inference-Time Domain Specialization

Kento Nishi
Harvard College

Madison Davis
Harvard College

Trevor DePodesta
Harvard SEAS

{kentonishi,madisondavis,tdepodesta}@{college,college,seas}.harvard.edu

Abstract

Specializing large language models typically requires separate checkpoints or routing mechanisms, duplicating parameters and fragmenting learned representations. Discrete diffusion language models offer a compositional alternative. Their iterative denoising process admits conditioning signals at each step, which may allow a single model to specialize to different domains through lightweight embeddings rather than separate weights. Building on our preliminary work that demonstrated this approach at fixed scale, we test its practical viability by training diffusion LMs from 120M to 1B parameters, each augmented with domain embeddings that steer generation toward mathematics or narrative at inference time. Autoregressive baselines at 120M serve as reference points. Three findings emerge: autoregressive models retain an edge over diffusion at 120M; embedding-guided generation consistently outperforms unguided joint training across all scales tested; and separate per-domain models exhibit a parabolic scaling curve with optimal performance around 500M parameters. Conditional models offer stable improvements over joint baselines without requiring scale-specific tuning, while separate training achieves the best absolute performance at intermediate scales but regresses at 1B.

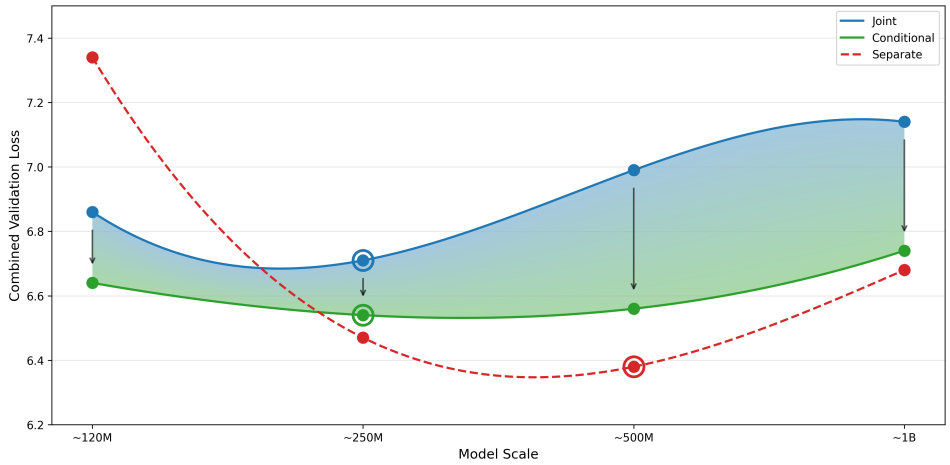

Figure 1: Scaling behavior of diffusion language models across three training strategies. Conditional models (green) consistently outperform joint training (blue) at all scales, with arrows indicating the gap between them. Separate per-domain models (red, dashed) exhibit a parabolic curve with optimal performance at 500M, suggesting a capacity sweet spot for domain-specific fine-tuning.

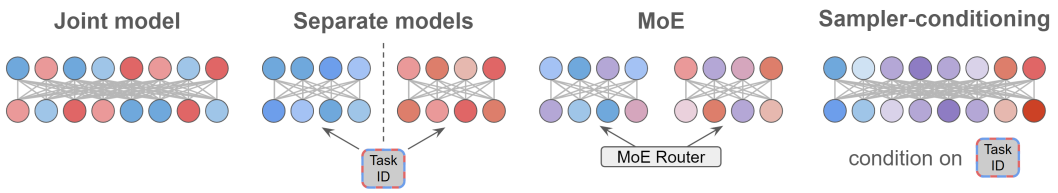

Figure 2: Compositional specialization: a single diffusion backbone plus domain embeddings may support task-specific generation without duplicating the model. Domain embeddings appear to bias the model toward math or narrative at inference time while keeping the backbone shared.

1 Introduction

1.1 Motivation

A recurring theme in AI is the tension between monolithic and compositional systems. Monolithic models learn end-to-end mappings that are powerful but opaque. Compositional systems decompose problems into reusable modules that can be recombined, offering interpretability and flexibility at the cost of integration complexity. Language model specialization sits squarely in this tension. Practitioners typically choose among three strategies: *multitask training*, which places all tasks in a single checkpoint and relies on prompts to select behavior; *expert routing*, which partitions capacity across subnetworks activated by a learned gate but introduces redundancy as each expert often re-learns common representations; and *per-domain fine-tuning*, which adapts separate models for each application but scales parameter counts linearly with the number of specializations. All three strategies encode task identity in the weights or in discrete routing decisions made during the forward pass. Autoregressive decoders are particularly constrained: once a token is emitted, generation cannot revise earlier choices based on later domain signals. An alternative approach might decompose behavior into domain-specific specializations that can be activated at inference time, though whether this offers practical advantages over existing methods remains an open question.

1.2 Approach

Discrete diffusion models offer a naturally compositional generation process: text is produced by progressively refining a corrupted input over many denoising steps, and because each step recomputes token probabilities conditioned on the current state, external signals—such as domain embeddings—can be injected throughout generation rather than only at the start. Following our preliminary work in CS2420 demonstrating this approach at 120M scale [Davis et al., 2025], we test the practical viability of diffusion-based LLMs by training variants up to 1B parameters.

Our key research question is: do any observed benefits of domain specialization in DLMs (i.e., learned per-domain embeddings that may specialize behavior without duplicating weights by biasing the denoising trajectory at inference-time) persists at scale, or do those advantages disappear as model size increases? We train diffusion LMs at 120M, 250M, 500M, and 1B parameters to observe how the effect of domain embeddings varies with model capacity. Our hope is that the idea of steering outputs toward task-appropriate distributions while keeping the backbone weights shared remains useful at scale.

1.3 Contributions

We present experiments spanning 120M to 1B parameters testing the empirical scalability of embedding-based domain conditioning in diffusion LMs (Figure 1). We observe how this approach performs relative to joint and separate-checkpoint baselines as model capacity grows. We replicate the preliminary 120M results from Davis et al. [2025] alongside controlled autoregressive baselines (joint, MoE, and separate configurations) under matched compute. Our experiments reveal distinct scaling behaviors: conditional models consistently outperform joint training at all scales, while separate per-domain models follow a parabolic trajectory with optimal performance around 500M. The underlying intuition, established in prior work, is straightforward: an unguided diffusion

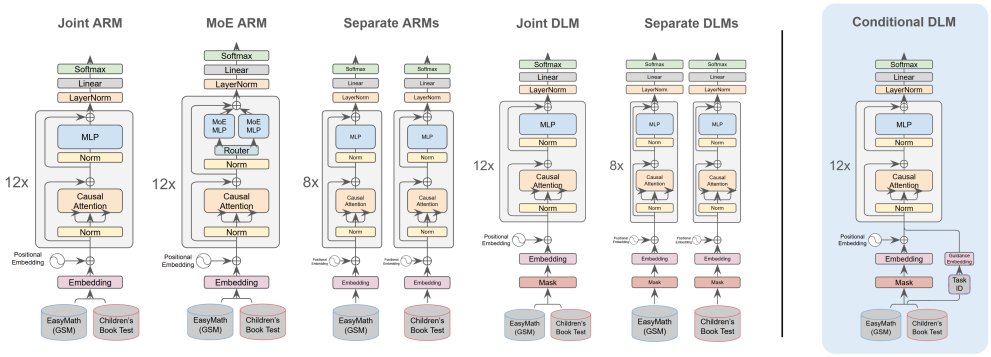

Figure 3: System overview (Joint ARM, MoE ARM, Separate ARMs, Joint DLM, Separate DLMs, Conditional DLM) with shared FineWeb-Edu pretraining and EasyMath/CBT fine-tuning. Separate variants train one model per domain.

model averages over all training domains when predicting masked tokens, while adding a domain embedding may focus each denoising step on a relevant subset of the data distribution.

2 Related Work

Diffusion-based language modeling has evolved rapidly over the past several years. Diffusion-LM [Li et al., 2022] introduced continuous-space denoising for controllable text generation, D3PM [Austin et al., 2021] extended the framework to discrete state spaces, and more recently LLaDA [Nie et al., 2024] demonstrated that discrete diffusion can match autoregressive perplexity at billion-parameter scales, hinting that the architecture gap may narrow with size. In image generation, classifier-free guidance [Ho and Salimans, 2022] showed that conditioning vectors can modulate each denoising step without an auxiliary classifier, enabling flexible control over generation. Applying this conditioning principle to text, our preliminary course project in CS2420 [Davis et al., 2025] introduced domain embeddings for steering diffusion language model generation toward task-specific behaviors, demonstrating the basic feasibility of the approach through six 120M-parameter model configurations under fixed compute. The present study extends that preliminary idea, testing whether the observed benefits were artifacts of the specific 120M scale or represent a more general property by comparing DLMs with 250M, 500M, and 1B parameters.

3 Method

3.1 Model Configurations

We evaluate six configurations at 120M parameters plus larger diffusion variants at 250M, 500M, and 1B (Figure 3). The key contrast is between non-compositional approaches (separate checkpoints and MoE routing) and compositional approaches that pair a shared backbone with domain embeddings. For the autoregressive baselines at 120M, Model 1 is a 12-layer GPT-2 variant (768-d, 12 heads, 1024 context) trained on both domains without explicit task tokens, Model 2 replaces feed-forward layers with mixture-of-expert blocks and a learned router, and Model 3 splits the budget into two 60M networks each fine-tuned on a single domain. On the diffusion side, Model 4 is a joint 12-layer transformer denoiser with 640-d embeddings and roughly 123M parameters, Model 5 mirrors the separate-AR design with two 60M diffusion networks serving as a non-compositional baseline, and Model 6 is our conditional variant that augments Model 4 with a two-entry embedding table containing only $\sim$1.3k additional parameters—selecting an embedding at inference composes the base model with domain-specific behavior. At larger scales (250M, 500M, 1B), we train Joint, Separate, and Conditional variants following the same template while scaling depth and width proportionally.

3.2 Compositional Denoising

We adopt the Block Diffusion framework [Arriola et al., 2025], which partitions sequences into length-16 blocks. Within each block a shared corruption rate t_b is sampled, and tokens are independently masked with probability t_b. The corrupted sequence

$$\mathbf{x}_t = \mathbf{m}_t \odot [\text{MASK}] + (1 - \mathbf{m}_t) \odot \mathbf{x}_0 \tag{1}$$

where $\mathbf{x}_0$ is the original clean sequence, $\mathbf{m}_t \in \{0, 1\}^n$ is a binary mask indicating which tokens are corrupted at noise level t, and $\mathbf{x}_t$ is the resulting partially-masked input. This corrupted sequence is concatenated with the clean suffix and fed to a transformer that predicts the original tokens at masked positions. Training minimizes a block-weighted cross-entropy objective,

$$\mathcal{L} = \mathbb{E}\left[\frac{\mathbf{m}_t}{t_b + \epsilon}\, \text{CE}\big(f_\theta(\mathbf{x}_t, t, \mathbf{e}_c),\, \mathbf{x}_0\big)\right], \tag{2}$$

where $\mathbf{e}_c$ is a learned domain embedding added to every token representation. The compositional structure is explicit: f_θ is the shared backbone, and $\mathbf{e}_c$ is a modular component that may specialize the model to a particular task. Setting $\mathbf{e}_c = \mathbf{0}$ recovers unguided generation, while setting it to the math or narrative vector biases denoising toward that domain, potentially decomposing multitask behavior into swappable task-specific signals—though the extent to which this represents true decomposition versus a form of prompt-like conditioning is unclear.

3.3 Data

All models are first pretrained on a 10B-token subset of FineWeb-Edu, then fine-tuned on two downstream tasks: grade-school math word problems in the style of GSM8K (8.9k examples) and Children's Book Test cloze passages (CBT, 10k examples). The former tests numerical reasoning, the latter tests narrative coherence. Both are publicly available on HuggingFace.

3.4 Optimization

We use AdamW with cosine learning-rate decay throughout. We pretrain for approximately 1k steps at a learning rate of 6×10^{-4} with dropout disabled. For fine-tuning, we use a fixed schedule of 250 steps at a reduced learning rate of 6×10^{-5} and dropout 0.3, applying the same settings for all model sizes. This uniformity is imposed by time and compute constraints and may limit convergence, especially for larger models. All experiments run on consumer hardware (NVIDIA RTX 4090 and RTX 3080 GPUs), and diffusion losses are computed directly on masked-token prediction.

4 Experiments

We report validation loss (lower is better) on held-out splits. Three comparisons structure our analysis. First, we replicate the preliminary 120M results of Davis et al. [2025], examining whether stronger specialization (progressing from joint training to MoE to separate checkpoints) helps autoregressive models, and comparing the compositional approach (shared backbone plus embeddings) against unguided or separate-checkpoint diffusion. Second, we test the practical scalability of the embedding-based approach by training larger diffusion models at 250M, 500M, and 1B parameters. Third, we observe how the relative performance of embedding-guided versus unguided diffusion changes across scales, though interpreting these trends requires caution given the limited number of scale points and domains tested.

Several practical considerations emerged during scaling experiments. Memory constraints on consumer GPUs required us to reduce sequence length and increase gradient accumulation for the 1B models. We also identified a normalization mismatch between training and evaluation phases. Different noise ranges were used during training versus validation, potentially impacting optimization dynamics and final performance. For all experiments, we used a fixed 250-step fine-tuning schedule and a constant learning rate across scales. This choice, made for tractability, may have limited convergence for larger models and complicates cross-scale comparison. Additionally, the separate-checkpoint baselines split capacity equally across domains, which may disadvantage them when domains have asymmetric complexity or data availability. The math task (GSM8K) contains fewer examples than the narrative task (CBT), yet each receives the same model capacity when trained separately.

Model	Arch	Type	Scale	Combined	CBT	GSM
AR Baselines (∼120M)						
1	AR	Joint	124M	1.86	3.09	0.645
2	AR	MoE	124M	1.80	2.97	0.646
3	AR	Separate	2×60M	1.75	2.84	0.66
DLMs (∼120M)						
4	DLM	Joint	123M	6.91	6.75	7.08
5	DLM	Separate	2×64M	7.34	7.14	7.53
6	DLM	Conditional	123M	6.64	6.58	6.70
DLMs (∼250M)						
4	DLM	Joint	264M	6.71	6.48	6.95
5	DLM	Separate	2×123M	6.47	6.35	6.59
6	DLM	Conditional	264M	6.54	6.38	6.71
DLMs (∼500M)						
4	DLM	Joint	522M	6.99	6.87	7.11
5	DLM	Separate	2×264M	6.38	6.37	6.38
6	DLM	Conditional	522M	6.56	6.47	6.64
DLMs (∼1B)						
4	DLM	Joint	947M	7.14	6.99	7.29
5	DLM	Separate	2×522M	6.68	6.64	6.72
6	DLM	Conditional	947M	6.74	6.63	6.85

Table 1: Validation losses across model scales.

5 Results

Table 1 reports validation losses across all configurations, and Figure 4 shows the validation loss curves of DLM variants across model scales. Several patterns emerge, though all warrant cautious interpretation given the practical constraints noted above.

At 120M parameters, autoregressive models dominate. Even the simplest AR model achieves a combined loss of 1.86, substantially outperforming the best diffusion variant at 6.64, replicating the preliminary findings. Within the AR family, adding MoE or splitting into per-domain networks yields incremental gains, suggesting that stronger specialization helps when capacity is limited. Within the diffusion family, the conditional variant (6.64) shows lower loss than both the joint baseline (6.91) and the separate-checkpoint approach (7.34). The embedding adds negligible parameters, roughly 1.3k. However, distinguishing genuine specialization from artifacts of the particular tasks, training dynamics, or the normalization issues identified during evaluation is difficult from this data alone.

We also examine whether these patterns persist as models grow. Comparing conditional versus joint diffusion across sizes, we observe gaps of 0.27 at 120M, 0.17 at 250M, 0.43 at 500M, and 0.40 at 1B. The gap remains positive at every scale, indicating that embedding-based conditioning consistently outperforms unguided joint training. Meanwhile, joint training exhibits a U-shaped curve: loss decreases from 120M to 250M, then increases at 500M and 1B. This suggests that larger joint models may struggle to balance multiple domains without explicit conditioning signals.

The separate-checkpoint baseline exhibits a striking parabolic scaling pattern. At 120M, separate training underperforms conditional (7.34 vs 6.64), likely because splitting capacity across two 64M models leaves each too small. Performance improves dramatically at 250M (6.47), crosses below conditional, and reaches a minimum at 500M (6.38)—the best result among all diffusion configurations. At 1B, however, separate training regresses to 6.68, suggesting an optimal scale around 500M for per-domain fine-tuning. This U-shaped curve may reflect a trade-off: smaller models lack sufficient capacity per domain, while larger models may overfit to their respective training sets or suffer from the overhead of maintaining entirely separate parameter sets.

Overall, the results reveal distinct scaling behaviors for each approach. Conditional models consistently outperform joint baselines at all scales, with stable performance that neither improves nor degrades substantially as capacity grows. Separate models follow a parabolic trajectory with an optimal scale around 500M, achieving the best absolute performance there but regressing at 1B. Joint models show the opposite pattern, with loss increasing at larger scales. These patterns suggest

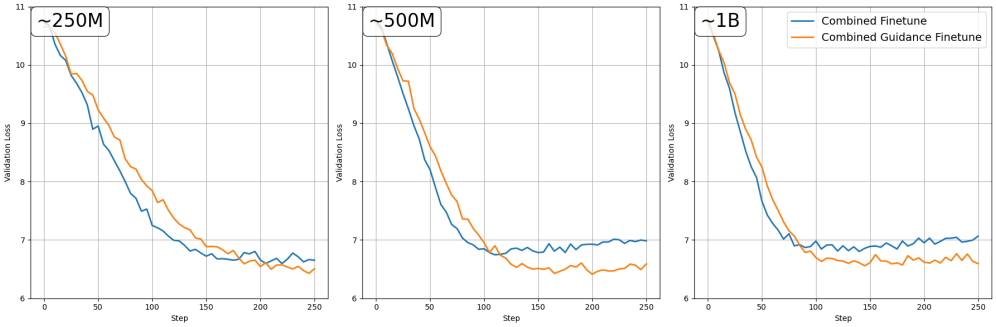

Figure 4: Validation losses for finetuned DLMs across model scales (models 4 and 6).

that embedding-based conditioning offers reliable improvements over joint training, while separate training may be preferable when a specific optimal scale can be identified.

6 Conclusion

We tested the practical scalability of compositional domain specialization in diffusion language models, extending preliminary 120M results to 250M, 500M, and 1B parameters. Autoregressive decoders substantially outperform diffusion at 120M, replicating earlier findings. Within diffusion, the conditional approach consistently outperforms joint baselines at all scales tested. Separate-checkpoint models exhibit a parabolic scaling curve, achieving optimal performance at 500M but regressing at 1B. Conditional models offer more stable performance across scales, suggesting that embedding-based conditioning provides reliable benefits without requiring scale-specific tuning. These results indicate that the compositional approach observed in preliminary work generalizes beyond 120M, though separate training may yield better absolute performance when the optimal scale can be identified.

6.1 Future Directions

Extensions include exploring interpolated or blended embeddings, testing on richer domain suites such as code and dialogue, and investigating whether the observed patterns generalize beyond the math and narrative tasks. The relationship between embedding-based conditioning and other forms of inference-time control also warrants further investigation.

References

Marianne Arriola, Zhanlin Qi, Aaron K. Gokaslan, Justin Han, Justin T. Chiu, Subham Sekhar Sahoo, Zhifeng Yang, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. arXiv preprint arXiv:2503.09573, 2025.

Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993, 2021.

M. Davis, K. Nishi, H. Zhang, and V. Cai. Sampler-conditioned diffusion language models. CS2420 course project, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chao Li. Large language diffusion models. arXiv preprint arXiv:2407.04907, 2024.