15 Research Frontiers in 5D Parallelism & Modern AI

Capstone Research Directions for the
GPU Engineer's Bootcamp

Vizuara AI Labs

March 2026

Scope. Each topic combines a **modern AI breakthrough** (diffusion LLMs, MoE scaling, reasoning models, RLHF, etc.) with a **5D parallelism dimension** (Tensor, Data, Pipeline, Context/Sequence, Expert Parallelism) to yield a novel research direction suitable for publication at **NeurIPS, ICML, or ICLR**.

5D Parallelism Dimensions:

- **TP** — Tensor Parallelism (split weight matrices across GPUs)
- **DP** — Data Parallelism (replicate model, split data batches)
- **PP** — Pipeline Parallelism (split model layers across stages)
- **CP/SP** — Context / Sequence Parallelism (split along sequence length)
- **EP** — Expert Parallelism (distribute MoE experts across GPUs)

Compiled with references to 50+ papers from 2024–2026.

Contents

1. Parallelism-Aware Training of Diffusion Language Models

Tensor Parallelism **Data Parallelism** **High Novelty**

Core Idea

Diffusion Language Models (DLMs) such as Mercury, MDLM, and SEDD generate *multiple tokens simultaneously* via iterative denoising, fundamentally breaking the sequential assumption of autoregressive models. Yet all existing parallelism strategies (Megatron-LM, DeepSpeed) are designed for autoregressive architectures. **This project designs tensor- and data-parallelism strategies that exploit the parallel token-generation structure of DLMs during both training and inference.**

Why It Is Impactful

- Mercury (2025) achieves 1,109 tokens/sec on H100s—10× faster than autoregressive models—but its training parallelism is under-explored.
- DLMs compute attention over *all* token positions at every denoising step (no causal mask), creating different communication patterns for TP than standard Transformers.
- A recent study (Jan 2026) showed MDLMs at 100B parameters still lag autoregressive models due to weak inter-token dependency modeling—better parallelism-aware architectures could close this gap.
- **No published work** co-designs TP/DP strategies for diffusion-based language model training.

Suggested Approach

Design custom all-reduce patterns that exploit the bidirectional attention structure of DLMs. Profile communication-to-computation ratios at each denoising step and adaptively switch between TP degrees. Benchmark against Megatron-LM baselines on GPT-scale DLMs.

Key References

Mercury (Inception Labs, 2025) • MDLM (NeurIPS 2024) • SEDD (ICML 2024) • "Parallelism and Generation Order in MDLMs" (Jan 2026)

Target: NeurIPS / ICML

2. Mixture-of-Experts Meets Diffusion LLMs: Expert Routing for Parallel Token Denoising

Expert Parallelism **Tensor Parallelism** **High Novelty**

Core Idea

In autoregressive MoE models, expert routing decides *which expert processes each token sequentially*. In Diffusion LLMs, *all tokens are denoised in parallel*, creating a fundamentally different routing problem. **This project designs MoE architectures for DLMs where expert selection is conditioned on both token identity and denoising timestep**, enabling timestep-aware expert specialization.

Why It Is Impactful

- No published work combines MoE with diffusion-based text generation—this is an entirely open design space.
- MoE reduces per-token FLOPs (only k of N experts fire), which is critical when DLMs process the *entire* sequence at every step.
- Expert Parallelism distributes experts across GPUs, but DLMs' all-to-all routing pattern (every token routes simultaneously) creates unique load-balancing challenges.
- Could yield the first trillion-parameter DLM that is trainable on current hardware.

Suggested Approach

Extend DeepSeek-style MoE routing with a timestep embedding input. Design load-balancing losses that account for the fact that early denoising steps (coarse structure) may need different experts than late steps (fine detail). Evaluate EP strategies with coarse-to-fine expert grouping.

Key References

MoE++ (ICLR 2025 Oral) • BigMac (AAAI 2025) • Mercury (2025) • Occult (NeurIPS 2025)

Target: ICLR / NeurIPS

3. Adaptive Context Parallelism with Dynamic Sparse Attention

Context Parallelism **Sequence Parallelism** **High Novelty**

Core Idea

Current context parallelism (CP) splits the sequence uniformly across GPUs, ignoring that attention is *highly sparse and non-uniform* across layers. **This project dynamically adjusts the CP degree per layer based on runtime attention sparsity patterns**, allocating more GPUs to dense-attention layers and fewer to sparse ones.

Why It Is Impactful

- MTraining (2025) showed that scaling dynamic sparse attention to distributed training suffers from worker-level and step-level imbalance, with actual speedups falling far below theoretical potential.
- Native Sparse Attention (NSA, DeepSeek 2025) demonstrates that different layers exhibit vastly different sparsity patterns ("Vertical-Slash" locality)—fixed CP wastes resources on already-sparse layers.
- RingX (SC25) achieves 3.4× speedup over Ring Attention but still uses uniform partitioning.
- Adaptive CP could save 30–50% communication for 1M+ token sequences.

Suggested Approach

Profile per-layer attention density during a warmup phase. Build a lightweight runtime scheduler that assigns CP degree per layer (e.g., CP=8 for dense early layers, CP=2 for sparse late layers). Integrate with Ring Attention's overlapped communication. Evaluate on 128K–1M context lengths.

Key References

MTraining (2025) • NSA (ACL 2025) • RingX (SC25) • Ring Attention (ICLR 2024) • Striped Attention (2024)

Target: NeurIPS / ICML

4. Speculative Decoding for Discrete Diffusion Models

Data Parallelism **Tensor Parallelism** **High Novelty**

Core Idea

Speculative decoding accelerates autoregressive inference by drafting tokens cheaply and verifying in parallel. **This project adapts speculative decoding to discrete diffusion LLMs**: a small "draft denoiser" predicts which tokens to unmask at each step, and a large "verifier denoiser" confirms or corrects in a single forward pass—across multiple GPUs.

Why It Is Impactful

- PEARL and DSI (both ICLR 2025) show 1.3–1.9× speedup for autoregressive speculative decoding, but no equivalent exists for DLMs.
- DLMs already generate tokens in parallel, but still require 10–100 denoising steps. Speculative "step skipping" could reduce this to 3–5 steps.
- EasySpec (Feb 2026) distributes draft model layers across GPUs—similar layer-parallel strategies apply to distributing denoising steps.
- Combines two of the hottest topics in ML systems: diffusion language models and speculative inference.

Suggested Approach

Train a lightweight draft denoiser (1/10th parameters) that predicts the token mask after k denoising steps. Run draft and verifier on separate GPU groups. Design acceptance criteria adapted to the discrete diffusion setting (edit-distance-based rather than token-probability-based). Benchmark latency reduction on code generation and open-ended text.

Key References

PEARL (ICLR 2025) • DSI (ICLR 2025) • EasySpec (Feb 2026) • "Divide and Conquer" for DLMs (Feb 2026)

Target: ICML / ICLR

5. Carbon-Aware Dynamic Parallelism Reconfiguration

All 5 Dimensions **High Novelty**

Core Idea

Current distributed training uses a *fixed* parallelism configuration (e.g., TP4-PP2-DP8) throughout the entire job. **This project builds a system that continuously monitors grid carbon intensity and dynamically reconfigures the parallelism strategy**—switching between TP/PP/DP/EP ratios—to minimize carbon footprint while maintaining throughput SLAs.

Why It Is Impactful

- Phantom Parallelism (2025) showed 50% energy reduction vs. standard TP, and GREEN (2025) demonstrated carbon-aware job scheduling—but *no work* reconfigures *parallelism dimensions* based on energy signals.
- Different parallelism strategies have dramatically different energy profiles: TP is communication-heavy (high NVLink power), PP has idle bubbles (wasted energy), DP is compute-dominated.
- Training a single frontier model emits 300+ tonnes of CO_2. Even a 15% reduction is significant.
- Highly interdisciplinary (systems + sustainability), which top venues increasingly value.

Suggested Approach

Profile energy consumption per parallelism configuration offline. Build a lightweight online controller that takes carbon-intensity signals (via WattTime/Electricity Maps API) and switches between pre-validated configurations at checkpoint boundaries. Prove convergence guarantees under configuration switching. Evaluate on LLaMA-scale training.

Key References

Phantom Parallelism (2025) • GREEN Framework (2025) • "Characterizing Distributed Training Efficiency" (MICRO 2025) • CarbonFlex (2025)

Target: NeurIPS / ICLR (Green AI Track)

6. Heterogeneous Expert Parallelism: Right-Sizing Experts to GPUs

Expert Parallelism **Data Parallelism**

Core Idea

Real-world GPU clusters are increasingly heterogeneous (H100 + A100 + L40S). Current MoE training assigns identically-sized experts uniformly. **This project introduces heterogeneous expert architectures where expert capacity (width/depth) is matched to the GPU it runs on**—larger experts on faster GPUs, smaller experts on slower ones—with a routing-aware load balancer.

Why It Is Impactful

- Cephalo (ICS 2025) and HetCCL (Jan 2026) demonstrated heterogeneous-cluster training, but both assume homogeneous model architecture. No work adapts the *model itself* to hardware heterogeneity.
- MoE models like DeepSeek-V3 use 256 identically-sized experts—placing them on mixed GPUs leaves faster GPUs idle waiting for slower ones.
- MoSE (Feb 2026) makes expert width adaptive for efficiency, but doesn't consider hardware heterogeneity.
- Could unlock MoE training for organizations that cannot afford homogeneous H100 clusters.

Suggested Approach

Design a "hardware-aware expert architecture search" that assigns expert hidden dimensions proportional to GPU FLOPS. Train with a modified load-balancing loss that targets equal *wall-clock time* (not equal tokens) per expert. Use profiling-guided expert placement. Evaluate on 2–3 heterogeneous cluster configurations.

Key References

Cephalo (ICS 2025) • HetCCL (Jan 2026) • MoSE (Feb 2026) • Hetis (Sep 2025) • MoE++ (ICLR 2025)

Target: NeurIPS / MLSys

7. Distributed Parallel Reasoning with Speculative Verification

Data Parallelism **Pipeline Parallelism**

Core Idea

Reasoning models (o1, DeepSeek-R1) generate extremely long chains-of-thought sequentially. **This project distributes reasoning across multiple GPUs**: a "planner" model decomposes problems into independent sub-tasks, multiple "executor" GPUs solve sub-tasks in parallel, and a "verifier" speculatively checks solutions before the full chain completes.

Why It Is Impactful

- SPRINT (NeurIPS 2025) demonstrated 39–65% sequential token reduction via parallel reasoning, but uses a single-GPU setup. *No work* distributes reasoning across a GPU cluster.
- Dynamic Parallel Tree Search (ACL 2025) parallelizes search but doesn't address distributed execution.
- Reasoning models produce 10K–100K tokens per query—distributed inference is essential for practical deployment.
- Combines inference-time compute scaling (a top research trend) with parallelism systems.

Suggested Approach

Fine-tune a planner to emit a DAG of sub-tasks with dependency annotations. Schedule independent sub-tasks across GPU workers using a DAG-aware scheduler with pipeline parallelism for the verifier. Implement speculative verification where the verifier begins checking partial results before all executors finish. Evaluate on math reasoning (MATH, GSM8K) and code generation (HumanEval).

Key References

SPRINT (NeurIPS 2025) • DPTS (ACL 2025) • SpecCoT (EMNLP 2025) • Continuous CoT (2025)

Target: ICLR / NeurIPS

8. Zero-Bubble Pipeline Parallelism for Diffusion Model Training

Pipeline Parallelism **Data Parallelism** **High Novelty**

Core Idea

Zero-bubble pipeline parallelism (ICLR 2024) eliminates bubbles by interleaving forward/backward passes—but assumes autoregressive models with standard backward passes. Diffusion models have *fundamentally different* training dynamics: loss is computed over the *entire* denoised sequence at every timestep, and gradients flow through a variable number of denoising steps. **This project designs pipeline schedules optimized for diffusion model training.**

Why It Is Impactful

- Zero-bubble PP and AdaPtis (Sep 2025) address autoregressive models. Diffusion training has different computation graphs that create novel bubble patterns.
- Image diffusion models (Stable Diffusion 3, FLUX) and video diffusion models (Sora) require massive compute. Better PP could reduce training cost by 20–30%.
- FreeRide (Middleware 2025) harvests bubbles for side tasks—diffusion training bubbles could be used for auxiliary denoising score estimation.
- Growing demand as diffusion models scale to billions of parameters.

Suggested Approach

Analyze the computation graph of diffusion training (forward denoising + score matching loss + backward). Design a "denoising-aware 1F1B" schedule that overlaps different timestep computations across pipeline stages. Implement on top of Megatron-LM. Evaluate bubble ratio and throughput on DiT and U-ViT architectures.

Key References

Zero Bubble PP (ICLR 2024) • AdaPtis (Sep 2025) • FreeRide (Middleware 2025) • Vocabulary Parallelism (Nov 2024)

Target: ICML / NeurIPS

9. Communication-Free Expert Parallelism via Learned Expert Compression

Expert Parallelism Tensor Parallelism

Core Idea

All-to-all communication in MoE models consumes 40%+ of training time at scale. MoE++ (ICLR 2025) introduced zero-computation experts for "easy" tokens. **This project goes further: train a lightweight autoencoder to compress the token representations sent between GPUs during expert dispatch**, reducing all-to-all communication volume by 4–8× without quality loss.

Why It Is Impactful

- BigMac (AAAI 2025) reduces communication by restructuring MoE to communicate at low dimensions (3× speedup), but requires architectural changes. A learned compressor is *architecture-agnostic*.
- Occult (NeurIPS 2025) optimizes expert co-location but cannot eliminate cross-node communication. Compression is complementary.
- MegaScale-MoE (ByteDance, 2025) overlaps all-to-all with compute but does not reduce the volume.
- Could enable MoE training on clusters with slower interconnects (e.g., Ethernet instead of InfiniBand).

Suggested Approach

Train a small bottleneck autoencoder (shared across all experts) that compresses token hidden states before dispatch and decompresses after receipt. Co-train the compressor with the MoE model using a reconstruction + task loss. Profile compression ratio vs. quality on a Pareto frontier. Test on DeepSeek-MoE-16B and Mixtral-8x7B.

Key References

MoE++ (ICLR 2025) • BigMac (AAAI 2025) • Occult (NeurIPS 2025) • MegaScale-MoE (2025)

Target: ICML / NeurIPS

10. Ring Attention + Speculative Decoding for Million-Token Inference

Context Parallelism Data Parallelism

Core Idea

Ring Attention distributes long-context KV-caches across GPUs for training, but *inference* with million-token contexts remains bottlenecked by sequential token generation. **This project combines Ring Attention's distributed KV-cache with speculative decoding**: draft tokens are generated locally while the ring asynchronously prefetches relevant KV-cache segments for verification.

Why It Is Impactful

- Meta's Context Parallelism (Nov 2024) extends ring attention to inference but doesn't address generation speed.
- Speculative decoding reduces latency but assumes the full KV-cache fits on one GPU—impossible for 1M+ token contexts.
- RingX (SC25) achieves 44% MFU on 4,096 GPUs for training but doesn't tackle inference.
- Long-context applications (document QA, code repository understanding) are a major deployment challenge.

Suggested Approach

Partition the KV-cache across GPUs in a ring topology. The draft model operates on a local KV-cache summary (compressed top-k keys). During verification, asynchronously gather full KV-cache segments from the ring for the draft token positions. Design a "speculate-then-gather" protocol that overlaps drafting with KV-cache retrieval. Evaluate latency on 128K–1M contexts.

Key References

Ring Attention (ICLR 2024) • RingX (SC25) • Context Parallelism for Inference (Meta, 2024) • PEARL (ICLR 2025)

Target: NeurIPS / ICLR

11. Asynchronous RLHF with Disaggregated Expert-Parallel Reward Models

Expert Parallelism Pipeline Parallelism

Core Idea

RLHF training jointly runs a policy model, reward model, reference model, and critic—spending 80% of time on generation. **This project uses MoE-based reward models with expert parallelism, disaggregated from the policy model's GPU pool**, enabling asynchronous reward scoring that overlaps with policy generation and training.

Why It Is Impactful

- Asynchronous RLHF (ICLR 2025) showed 40% speedup by decoupling generation and learning, but uses monolithic reward models.
- StreamRL (2025) disaggregates training and generation but doesn't optimize the reward model itself.
- MoE reward models could score different aspects of quality (helpfulness, safety, style) via different experts—each on different GPUs.
- OPPO (Sep 2025) overlaps PPO phases but doesn't address reward model parallelism.

Suggested Approach

Replace the reward model with an MoE architecture where experts specialize on different reward dimensions. Deploy reward experts across a separate GPU pool with expert parallelism. Design an asynchronous scoring protocol where reward computation happens in parallel with policy generation. Evaluate throughput and alignment quality on standard RLHF benchmarks (TL;DR summarization, HH-RLHF).

Key References

Async RLHF (ICLR 2025) • StreamRL (2025) • RLHFuse (2024) • OPPO (Sep 2025) • OpenRLHF

Target: ICML / NeurIPS

12. Scaling Test-Time Training Layers via Context Parallelism

Context Parallelism **Sequence Parallelism**

Core Idea

Test-Time Training (TTT) layers update model weights *at inference time* using the input as training data, enabling unbounded context. However, existing TTT methods achieve <5% GPU utilization due to tiny mini-batches. **This project scales TTT layers to million-token contexts using context parallelism with large-chunk gradient aggregation across GPUs.**

Why It Is Impactful

- LaCT (ICLR 2026) showed that large-chunk TTT achieves orders-of-magnitude better hardware utilization, and context parallelism enables distributed gradient aggregation with only 1–3% overhead.
- TTT layers offer a fundamentally different approach to long context than attention-based methods—they have *linear* complexity.
- Combining TTT with existing Transformer parallelism strategies is unexplored: how should TP (for attention layers) interact with CP (for TTT layers) within the same model?
- Could enable real-time adaptation to user-specific data at deployment scale.

Suggested Approach

Design a hybrid parallelism strategy: TP for attention layers, CP for TTT layers, with automatic switching at layer boundaries. Optimize the all-reduce-sum for TTT gradient aggregation across context shards. Benchmark on long-context tasks (RULER, Needle-in-a-Haystack) at 256K–1M tokens. Compare throughput and quality vs. pure attention models with Ring Attention.

Key References

LaCT / TTT Done Right (ICLR 2026) • TTT Layers (2024) • Ring Attention (ICLR 2024) • Scaling LLM Test-Time Compute (ICLR 2025)

Target: ICLR / NeurIPS

13. Topology-Aware Expert Placement for Disaggregated MoE Serving

Expert Parallelism **Pipeline Parallelism**

Core Idea

In MoE inference, frequently co-activated experts should be co-located on the same node to minimize cross-node all-to-all traffic. **This project learns a topology-aware expert placement policy from production routing traces**, using graph partitioning on the expert co-activation graph to minimize cross-node communication while maintaining load balance.

Why It Is Impactful

- MegaScale-Infer (SIGCOMM 2025) disaggregates attention and FFN modules but uses static expert placement.
- JANUS (Dec 2025) disaggregates attention and experts but doesn't optimize *which* experts go *where*.
- DeepSeek-V3's 256 experts create $\binom{256}{8} \approx 4.4$B possible routing combinations—intelligent placement is essential.
- Lmsys deployed DeepSeek with EP on 96 H100s, reporting that expert placement significantly impacts throughput.

Suggested Approach

Collect expert co-activation statistics from production traffic. Formulate placement as a graph partitioning problem: nodes = experts, edge weights = co-activation frequency, partitions = physical GPU nodes. Solve with spectral clustering + refinement under load-balance constraints. Design an online re-placement strategy for distribution drift. Evaluate on DeepSeek-V3 and Mixtral serving workloads.

Key References

MegaScale-Infer (SIGCOMM 2025) • JANUS (Dec 2025) • Lmsys DeepSeek deployment (2025) • Occult (NeurIPS 2025)

Target: OSDI / NeurIPS

14. Energy-Aware Expert Routing in Mixture-of-Experts Models

Expert Parallelism **Data Parallelism** **High Novelty**

Core Idea

Standard MoE routers select experts purely based on token-expert affinity scores. **This project adds an energy cost term to the routing decision**: tokens are routed considering both quality (affinity score) and the real-time energy cost of the GPU hosting each expert, creating energy-aware expert routing that naturally balances quality and sustainability.

Why It Is Impactful

- No existing work incorporates energy signals into MoE routing—this is a completely novel formulation.
- Modern GPU clusters have heterogeneous power states: some GPUs run at higher frequencies (more energy, faster), others in power-saving mode.
- MoE++ showed that "easy" tokens can use zero-computation experts. Energy-aware routing generalizes this: easy tokens go to low-power GPUs, hard tokens go to high-power GPUs.
- Could reduce inference energy by 20–40% with minimal quality degradation.
- Aligns with the growing "Green AI" movement at top venues.

Suggested Approach

Augment the router's gating function with a per-expert energy penalty: $g_i = \text{softmax}(\mathbf{x}^\top \mathbf{w}_i - \lambda \cdot e_i)$, where e_i is the real-time energy cost of expert i's GPU. Train with a multi-objective loss balancing task quality and total energy. Use DVFS (dynamic voltage/frequency scaling) readings as the energy signal. Evaluate quality-energy Pareto frontiers on standard benchmarks.

Key References

MoE++ (ICLR 2025) • Phantom Parallelism (2025) • "Energy Consumption in Parallel NN Training" (Aug 2025) • MICRO 2025 Thermal Study

Target: NeurIPS / ICLR (Green AI)

15. Unified Parallelism Search: AutoML for 5D Parallelism Configuration

All 5 Dimensions **High Novelty**

Core Idea

Choosing the right combination of TP, DP, PP, CP, and EP degrees is currently done by expert engineers via trial-and-error. **This project formulates 5D parallelism configuration as a combinatorial optimization problem and solves it with a learned cost model + search algorithm** that predicts throughput, memory usage, and communication cost for any configuration, then searches the 5D space efficiently.

Why It Is Impactful

- The 5D parallelism search space is enormous: for a 256-GPU cluster, there are thousands of valid (TP, DP, PP, CP, EP) configurations. Manual tuning leaves significant performance on the table.
- Meta showed that TP4-EP4-PP4 delivers 3× more throughput than TP64 for inference, but this was found via expert knowledge, not systematic search.
- Alpa (OSDI 2022) searched 2D (TP×DP) space. Extending to 5D with modern models (MoE, long-context, heterogeneous) is a major leap.
- Could become the "NAS for parallelism"—a foundational tool for the field.

Suggested Approach

Build a differentiable cost model that predicts per-iteration time given (TP, DP, PP, CP, EP, model config, hardware config). Train the cost model on profiling data from diverse configurations. Use Bayesian optimization or evolutionary search over the 5D space with the cost model as surrogate. Validate on 3+ model architectures (dense LLM, MoE, long-context) across 64–512 GPUs. Open-source the framework.

Key References

Alpa (OSDI 2022) • Meta Multi-Dim Parallelism (2025) • Megatron-LM • DeepSpeed • AdaPtis (Sep 2025)

Target: NeurIPS / OSDI / MLSys

Summary of All 15 Research Topics

#	Topic	Parallelism	Modern AI Concept
1	Parallelism-Aware DLM Training	TP + DP	Diffusion LLMs
2	MoE $\times$ Diffusion LLMs	EP + TP	DLM + MoE
3	Adaptive CP for Sparse Attention	CP + SP	Sparse Attention
4	Speculative Decoding for DLMs	DP + TP	Diffusion + Speculation
5	Carbon-Aware Parallelism Reconfig	All 5D	Green AI
6	Heterogeneous Expert Parallelism	EP + DP	Heterogeneous Clusters
7	Distributed Parallel Reasoning	DP + PP	Reasoning Models (o1)
8	Zero-Bubble PP for Diffusion Training	PP + DP	Diffusion Models
9	Communication-Free EP via Compression	EP + TP	Learned Compression
10	Ring Attention + Speculative Decoding	CP + DP	Long-Context Inference
11	Async RLHF with MoE Reward Models	EP + PP	RLHF
12	Scaling TTT Layers via CP	CP + SP	Test-Time Training
13	Topology-Aware Expert Placement	EP + PP	Disaggregated Serving
14	Energy-Aware Expert Routing	EP + DP	Green AI + MoE
15	AutoML for 5D Parallelism Config	All 5D	Neural Search / AutoML

How to Use This Document. Each topic is designed as a **standalone 3–6 month research project** suitable for a masters thesis, Ph.D. qualifying project, or conference paper. Topics marked **High Novelty** represent directions with *zero existing published work* at the time of writing (March 2026). We recommend starting with a focused literature survey, implementing a minimal prototype on 4–8 GPUs, and scaling experiments to validate the core hypothesis.