Transformer-Based SLAM Learning Checklist (PyTorch + CUDA)

Phase 1 – PyTorch Foundations (Weeks 1–3)

Goal: Master PyTorch, data loading, and attention basics.

☐ Complete PyTorch Tutorials.
☐ Implement Linear Regression & CNN manually.
☐ Inspect graphs with `torchsummary` or `torchviz`.
☐ Write custom `Dataset` & `DataLoader` using Data Loading Tutorial.
☐ Implement toy Vision Transformer (ViT) using ViT PyTorch (lucidrains).
☐ Visualize attention maps.

✅Deliverables: - `train_vit_toy.py` - Attention visualization notebook

Phase 2 – Transformer Feature Matching (Weeks 4–6)

Goal: Implement LoFTR-style matching on paired images.

☐ Read LoFTR Paper.
☐ Clone & study Official LoFTR Repo.
☐ Implement coarse feature matching (shared CNN + cross-attention).
☐ Train on KITTI or TUM-RGBD pairs.
☐ Use Kornia Matching Tutorial.
☐ Train using mixed precision (AMP Recipe).

✅Deliverables: - `loftr_simplified.py` - Heatmaps + validation loss plots

Phase 3 – Pose Estimation & CUDA (Weeks 7–9)

Goal: Estimate pose and optimize model using CUDA.

☐ Extract top-N matches from Transformer output.
☐ Estimate pose using `cv2.findEssentialMat` + `cv2.recoverPose` (OpenCV Docs).
☐ Evaluate translation/rotation errors.
☐ Profile using `torch.profiler` & Nsight Systems.
☐ Implement one custom CUDA kernel (Tutorial).
☐ Run ablations (embedding dim, attention heads, etc.).

✅Deliverables: - `pose_from_correspondences.py` - Pose trajectory + GPU profile logs

Phase 4 – Planning Extension (Weeks 10–12)

Goal: Extend Transformer pipeline to planning or trajectory prediction.

☐ Replace image inputs with state sequences `[x, y, θ, v, ω]`.
☐ Train Decision Transformer or Diffusion Planner variant.
☐ Study Decision Transformer Paper.
☐ Use Decision Transformer Repo.
☐ Try Diffusion Policy Repo.
☐ Compare perception vs. planning data format in a notebook.

✅Deliverables: - `transformer_planner.py` - Comparative visualization notebook

Final Outputs

☐ Fully working Transformer SLAM pipeline
☐ CUDA kernel + profiling results
☐ Extended planning pipeline
☐ Portfolio-ready documentation

Core Tools Summary

Category	Tools
Deep Learning	PyTorch, torchvision, Kornia
CUDA & Profiling	torch.profiler, Nsight Systems
Vision & Geometry	OpenCV, numpy
Datasets	KITTI, TUM-RGBD
Visualization	matplotlib, evo
Environment	Ubuntu + Conda (GPU RTX 3060+ recommended)

This checklist ensures week-by-week tangible progress from PyTorch mastery → Transformer SLAM → CUDA optimization → Transformer-based planning.