

# Generative AI Engineering – Scope of Material

The specialization develops practical competencies in artificial intelligence, machine learning, deep learning, and generative systems. The scope below presents the planned direction of the specialization and may be refined before classes begin or during the semester. The detailed scope, level of difficulty, and selection of projects may be adjusted to the size, preparation, and progress of the group. The program may also be modified in response to technological development and organizational requirements.

The emphasis is on understanding methods, implementing them in Python, and building solutions that can be demonstrated, tested, and further developed as part of an engineering thesis. The program has an engineering and project-oriented character. Students will work with tools used in AI/ML practice, including NumPy, PyTorch, TensorFlow/Keras, FastAPI, Docker, MLflow, CI/CD, and Kubernetes. The goal of the specialization is not only to learn models, but also to build functioning, testable, and responsibly deployed AI systems. The specialization prepares students for the expectations of the IT market.

## UGP – Deep Learning in Python

Organizing the foundations of DL and preparing students for more advanced AI projects, including RAG, RL, and systems based on deep models.

| Scope | Topic                           | Key elements                                                                        |
|-------|---------------------------------|-------------------------------------------------------------------------------------|
| 1     | Foundations of neural networks  | Perceptron, vectors and matrices in NumPy, classification, basic metrics.           |
| 2     | Multi-layer networks            | MLP, forward and backward propagation, Softmax, TensorFlow/Keras and PyTorch.       |
| 3     | Optimization and regularization | Loss functions, SGD/Adam, overfitting, callbacks, learning curves.                  |
| 4     | CNNs and images                 | Convolution, pooling, MNIST/Fashion-MNIST/CIFAR classification, transfer learning.  |
| 5     | Detection and temporal models   | Object detection, tracking, simple video processing, pipeline optimization.         |
| 6-7   | Autoencoders, VAE and GAN       | Latent space, data generation, training stability, comparison of VAE and GAN.       |
| 8     | RNN, LSTM and GRU               | Sequences, vanishing gradient problem, time-series prediction, Stooq data.          |
| 9-11  | RL and Deep RL                  | Q-learning, MDP, exploration/exploitation, Deep Q-learning, robotics examples.      |
| 12-13 | Attention, Transformers and RAG | Embeddings, vector search, LLM API integration, building a simple RAG pipeline.     |
| 14-15 | Ethics, law and AI limitations  | Privacy, copyright, bias, model limitations, computational cost, future directions. |

Projects: CIFAR-100 image classification, implementation of RL/Deep RL, and a simple LLM+RAG system with embeddings and vector search.

## PRO1 – Project: AI Systems Engineering and MLOps

Transition from an ML/DL model to a working service that can be run locally, tested, documented, and presented.

| Scope | Topic                  | Key elements                                                                                                           |
|-------|------------------------|------------------------------------------------------------------------------------------------------------------------|
| 1-3   | Model API              | FastAPI, REST, Pydantic, /predict endpoint, input/output validation, error handling, logs and metrics.                 |
| 4-5   | Containerization       | Docker, Dockerfile, environment configuration, Docker Compose, auxiliary services, volumes, and basics of API traffic. |
| 6-8   | Data and preprocessing | Data versioning, schemas, train/val/test split, quality validation, robustness to incorrect or corrupted input.        |
| 9-10  | Model lifecycle        | MLflow Tracking, artifacts, model registry, experiment comparison, basic quality tests and smoke tests.                |
| 11    | CI/CD and monitoring   | GitHub Actions, lint/test/build, basic drift detection and monitoring of input or embedding statistics.                |
| 12    | Inference performance  | Latency, throughput, cache, batching, warmup, basics of quantization, and measuring the operating cost of the service. |
| 13    | Kubernetes             | Pod, Deployment, Service, ConfigMap, readiness/liveness probes, local deployment in kind/minikube.                     |
| 14    | RAG as a service       | API + vector database, index initialization, /ask endpoint, query tests and system limitations.                        |
| 15    | System finalization    | Swagger/OpenAPI, error standardization, request-id, deployment checklist, project demo.                                |

Projects: A local AI system, for example an API for a CIFAR model or RAG, with preprocessing, data validation, Docker/Docker Compose, MLflow, metrics, tests, and documentation.

## GAI – Generative AI

Study of modern generative and multimodal systems from the perspectives of architecture, implementation, optimization, evaluation, and deployment.

| Scope | Topic                                  | Key elements                                                                                                            |
|-------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 1     | Vision Transformers                    | Image patches, ViT block, CNN vs ViT comparison, influence of patch size.                                               |
| 2     | Diffusion models                       | Adding noise, denoising, 1D examples, number of sampling steps vs quality.                                              |
| 3-5   | Multimodality and CLIP                 | Text-image embeddings, cross-attention, contrastive learning, image-text retrieval, prompt sensitivity.                 |
| 6     | GNN                                    | Graphs, GCN layer, message passing, node classification, comparison with MLP.                                           |
| 7     | RAG                                    | Vector database, retrieval, context for LLM, factuality measurement and error analysis.                                 |
| 8     | Preference alignment                   | RLHF, DPO, preference data, reward model, reward model overfitting.                                                     |
| 9     | Efficient fine-tuning                  | LoRA, adapters, freezing weights, number of parameters, GPU memory usage.                                               |
| 10    | Model compression                      | Distillation, quantization, pruning, inference latency, quality degradation.                                            |
| 11-12 | Model scaling                          | MoE, expert routing, load balancing, distributed training, batch size, throughput.                                      |
| 13-15 | Deployment, agents and structured data | Guardrails, robustness, red-teaming, tool-using agents, interpretability, structured outputs and robotics applications. |

Projects: Multimodal image-text retrieval in a CLIP-like style and an efficient LLM/RAG system with LoRA, quantization, and analysis of the quality-time-memory trade-off.

## PRO2 and PRO3 – Project and Presentation Classes

PRO2. These classes have a lecture and presentation format and are intended to organize the broader context of generative AI development. Topics include scaling laws, computational cost, frontier models, multimodal models, AI in science, medicine, robotics, creative industries, and professional work, as well as hallucinations, reliability, synthetic data, safety, alignment, and possible directions beyond Transformers. The goal is to develop the ability to critically assess trends, not only to learn a single tool.

PRO3. Classes in the diploma semester support preparation of the engineering thesis and final presentation. The scope includes thesis structure, deadlines and formal requirements, LaTeX/Overleaf or equivalent tools, formatting tables, figures, code and bibliography, academic writing style, PJATK requirements, plagiarism issues, consultations, review of thesis fragments, oral presentations, and mock preparation for the defense.

The program may be subject to change. Publication date: 30.05.2026.

Coordinator of specialization M: Dr. Eng. Michał Majewski