

Generative AI-Based Design of Electric Machines

A CVAE-based Framework for the Optimisation of Interior
Permanent Magnet Motors for the Automotive Market

Lorenz De Maeyer

Thesis submitted for the degree of
Master of Science in Engineering:
Energy

Supervisors

Prof. dr. ir. R. V. Sabariego

Dr. ir. I. Siera

Assessors

Prof. dr. ir. H. Kazmi

Ir. T. Verschelde

Assistant-supervisor

Ir. S. Toghranegar

© 2026 KU Leuven – Faculty of Engineering Science

Published by Lorenz De Maeyer,

Faculty of Engineering Science, Kasteelpark Arenberg 1 bus 2200, B-3001 Leuven

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher. This publication contains the study work of a student in the context of the academic training and assessment. After this assessment no correction of the study work took place.

Preface

This thesis marks the end of my studies at KU Leuven and would not have been possible without the guidance and support of several people, to whom I would like to express my sincere appreciation.

I would first like to thank my supervisors, Prof. Ruth V. Sabariego and Dr. Ilja Siera, for entrusting me with this topic and for their guidance throughout the project. Ilja, your mentorship and practical insights gave this work an industrial grounding that no textbook could have offered.

I am also particularly grateful to my assistant, Sina Toghranegar. Sina, your expertise on the machine-learning side and your patience during our weekly discussions kept the project on track.

Their support has been fundamental to the completion of this thesis.

Lastly, I thank the jury members for their time and effort in reviewing this thesis, and my family and friends for their support throughout my studies.

Lorenz De Maeyer

Contents

Preface	i
Abstract	iv
List of Figures and Tables	v
List of abbreviations and symbols	xi
1 Introduction	1
1.1 Industrial Context: Electric Motor Development at Punch Powertrain	1
1.2 Problem Statement and Research Objectives	7
2 Theoretical Background	9
2.1 The Interior Permanent Magnet Synchronous Machine	9
2.2 Artificial Neural Networks	16
2.3 Deep Generative Models	17
2.4 Network Design and Training Techniques	23
2.5 AI-Based Electric Machine Design	23
3 Dataset Construction and Performance Abstraction	25
3.1 Data Generation Pipeline	25
3.2 Operating Points and Condition Vector	28
3.3 Dataset and Preprocessing	30
4 Surrogate MLP for Geometry-to-Performance Mapping	35
4.1 Network Inputs, Outputs and Architecture	35
4.2 Output-space Compression by Principal Components	37
4.3 Training Configuration and Convergence	38
4.4 Test-set Accuracy on the Denormalised Performance Profile	39
4.5 Deployment within the Framework	42
5 Conditional Variational Autoencoder	43
5.1 Model Architecture	43
5.2 Standard CVAE	45
5.3 Performance-Regularised CVAE	46
5.4 Inference Procedure	47
5.5 Evaluation	48
6 Latent Space Optimisation	57
6.1 Problem Formulation	57

6.2	Gradient-Based Method	58
6.3	Results	60
6.4	Discussion	64
7	Conclusion	65
7.1	Summary	65
7.2	Main Findings	66
7.3	Limitations and Outlook	67
A	Supplementary Theoretical Background	71
A.1	Formal Multi-Objective Optimisation Formulation	71
A.2	The ZONDA Design Computation Platform in Detail	72
A.3	The Arbitrary Reference Frame: Clarke and Park Transform	73
A.4	Winding Ratio: Technical Detail	74
A.5	Neural-Network Training Techniques and Hyperparameter Detail	75
A.6	Variational Autoencoder: Full Derivation	79
B	Dataset and Preprocessing	81
B.1	Representative Dataset Rows	81
B.2	Supplementary EDA Figures	82
C	CVAE Evaluation Details	85
C.1	Per-feature reconstruction metrics	85
C.2	Latent-space diagnostics	87
C.3	Alternative reconstruction training objectives	92
C.4	Case-study details	101
D	Latent-Space Optimisation Results	107
	Bibliography	119

Abstract

Electric traction motors for hybrid and battery vehicles are currently designed by wrapping a finite-element key-performance-indicator (KPI) evaluator in an evolutionary optimisation loop, an approach that can take days of compute on a multi-core node per customer case. This thesis addresses the same problem from the inverse direction: given a target peak performance profile together with the electrical limits imposed by the inverter and battery, generate a diverse set of compliant double-V interior permanent magnet synchronous machine (IPMSM) geometries. The proposed framework combines three components, using a dataset of 85 384 physically valid samples obtained by latin hypercube sampling of the 17 free geometric parameters: (i) a residual multilayer-perceptron surrogate that maps geometry and electrical limits to the 80-point peak torque–speed curve through a 19-component principal-component bottleneck, (ii) a conditional variational autoencoder (CVAE) with a 15-dimensional latent space that learns to generate performance-compliant geometries, given the working conditions, and (iii) a gradient-based latent space optimisation stage that refines a sampled candidate towards minimal volume under soft performance-compliance and structural-integrity penalties driven by the frozen surrogate.

On the held-out test set, the surrogate reaches a mean relative L_2 error of 0.62 % on the denormalised torque curve at sub-millisecond inference cost. A performance-regularised CVAE variant, which augments the evidence lower bound with a differentiable compliance penalty derived from the frozen surrogate, lifts the mean prior-sampling compliance on eight industrial conditions from 49.1 % to 75.7 % while preserving roughly 86 % of the per-feature design-range coverage. The subsequent latent-space optimisation returns a compliant candidate for every one of the eight conditions and for both CVAE variants, converging within roughly 100 iterations and finishing in about 45 s per condition on a single CPU core. A single FEM run confirms that the optimised torque–speed curve passes through the requested operating points. The framework therefore compresses what is currently a days-long search into a sub-minute generative step. Thermal, efficiency, noise-vibration-harshness, and demagnetisation behaviour remain out of scope and are identified as the priority directions for further work.

List of Figures and Tables

List of Figures

1.1	Vehicle-level torque–speed and power–speed requirement characteristics, as typically specified by the client at the wheels, showing peak and continuous operating envelopes. Source: Punch Powertrain	3
1.2	Closed-loop optimisation between machine parameters and KPI evaluation via the Design Computation Platform. Source: Punch Powertrain	4
1.3	Two valid double-V IPMSM rotor designs and their corresponding parametrizations.	5
1.4	KPI calculation engine.	6
2.1	Double V-shaped IPMSM design with 4 poles. Source: Punch Powertrain	9
2.2	Three common rotor topologies in the IPMSM family.	10
2.3	Synchronous (rotor-locked) reference frame for the double-V IPMSM. .	11
2.4	Operating modes of an IPMSM in the dq current plane, along the maximum-torque trajectory A–B–C–D. Reproduced from [34].	13
2.5	Conventional data generation workflow, used in the ZONDA Toolchain.	13
2.6	Flux map generation and resulting d- and q-axis flux maps.	14
2.7	Schematic of a feedforward multilayer perceptron with one hidden layer.	16
2.8	Illustration of the reparameterisation trick [25]. Original (left) and reparameterised (right) computational graphs for $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x})$	20
2.9	Training-time architectures of (a) the VAE and (b) the CVAE. In both cases the encoder maps the input to a Gaussian posterior with parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma})$, the latent $\mathbf{z}$ is obtained by the reparameterisation trick (Figure 2.8), and the decoder reconstructs $\hat{\mathbf{x}}$	21
2.10	CVAE at inference time. The encoder of Figure 2.9(b) is dropped: $\mathbf{c}$ is fixed at the design requirement and K prior samples $\mathbf{z}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ drive the decoder, yielding K diverse candidate geometries in one forward pass each.	22
3.1	Data-generation workflow used in this work. Each iteration produces one $(\mathbf{G}, I_{max}, U_{max}, N, PP)$ data sample.	27

3.2	Reduction of the full performance profile to the two operating points OP1 and OP3.	28
3.3	Compliance through two-point evaluation. A sample is accepted iff its computed PP lies above the requirement at both ω_{OP1} and ω_{OP3}	29
3.4	Distribution of OP1 and OP3 in the torque–speed plane. Left: exploratory 10^3 -sample run (845 post-filter). Right: working dataset (85 384 post-filter samples).	31
3.5	Pair-matrix of the condition vector $\mathbf{C}$ and the three geometry parameters most strongly correlated with $\mathbf{C}$ (KAR, DSO, KMQ1).	32
4.1	Network architecture of the surrogate MLP. Blue blocks are trainable; the green (z -score) and orange (PCA^{-1}) blocks are deterministic, fitted-once transformations exported alongside the network weights.	36
4.1	Network architecture of the surrogate MLP (continued).	37
4.2	Characterisation of the output-space PCA.	38
4.3	Training (blue) and validation (orange) Huber loss versus epoch. Vertical dashed lines mark learning-rate reductions triggered by the plateau scheduler; the run is terminated by the early-stopping criterion shortly after the best validation loss is reached.	39
4.4	Parity plot of predicted versus reference torque over the entire test set, aggregated across all 80 RPM samples per curve.	41
4.5	Distribution of the per-sample relative L_2 error (4.7) on the test set. Vertical lines mark the mean, the median and the 95 th percentile.	41
4.6	Four randomly chosen test samples. For each sample, the raw ZONDA reference (solid), the PCA reference (dashed) and the surrogate prediction (dash-dot) are overlaid. The gap between solid and dashed is the irreducible PCA reconstruction error of §4.2; the gap between dashed and dash-dot is the residual attributable to the network itself.	42
5.1	Network architecture of the CVAE. Blue blocks are trainable; the lighter shade marks the deterministic sigmoid output. Dimension labels above each arrow indicate the tensor width at that stage, and concatenation of two tensors is denoted by $\oplus$	44
5.2	Training scheme of the standard CVAE.	45
5.3	Training scheme of the CVAE with extra performance regulariser.	46
5.4	Parity plots of decoded vs. target geometry for the most condition-sensitive features (KAR, DSO, KMQ1) of §3.3.1. Standard and regularised CVAE are overlaid in distinct colours; the diagonal denotes perfect reconstruction.	49
5.5	Per-dimension KL divergence on the test set for the standard (left) and regularised (right) CVAE.	49
5.6	Latent traversals along the top-3 most active dimensions of the standard and regularised CVAE.	50

5.7	Torque–speed curves of 100 candidate geometries generated for 4 of the 8 industrial conditions, evaluated through the MLP surrogate. Compliant samples are coloured green, non-compliant samples red; target operating points OP1 and OP3 are marked.	52
5.8	Marginal distributions of the 1000 generated samples for the first two industrial conditions, both model variants.	54
5.9	Aggregate Spearman split-correlation matrix of the predicted geometric features, averaged over the 8 industrial conditions. Lower triangle: standard CVAE; upper triangle: performance-regularised CVAE.	55
6.1	Gradient-based latent-space optimisation loop. The dashed red arrow indicates the gradient update on $\mathbf{z}$; all networks remain frozen.	59
6.2	Convergence curves for the 25 multi-start runs of Condition 1; the winner is highlighted.	61
6.3	Volume comparison for Condition 1: the 25 multi-start endpoints (green; winner in red) against 100 random CVAE samples (grey: all, blue: compliant). Standard (top) and regularised (bottom) CVAE. . . .	62
6.4	Validation of the winner for Condition 1: MLP-predicted torque–speed curve versus FEM ground truth, with OP1 and OP3 marked. Standard (top) and regularised (bottom) CVAE.	63
6.5	One-pole slice of the winning geometry for Condition 1. Standard (left) and Regularised (right) CVAE.	63
A.1	Design Computation Platform process: a geometry $\mathbf{G}$ is mapped through a flux-map FEM step, a winding-ratio scaling, and an MTPA/field-weakening construction to a peak performance profile, subject to the electrical limits of the client.	73
B.1	Spearman correlation between each condition (rows) and each geometry parameter (columns) on the post-filter dataset ($M = 85\,384$). The three columns most strongly correlated with $\mathbf{C}$ (KAR, DSO, KMQ1) are selected for the pair-matrix in Figure 3.5.	82
B.2	Pair-matrix between all 17 geometry parameters; cell conventions are identical to Figure 3.5. The absence of any significant off-diagonal correlation ($ \rho_S < 0.05$ throughout) confirms that the LHS marginals remain effectively independent after feasibility filtering.	83
C.1	Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and regularised CVAE.	86
C.2	Marginal posteriors of the 15 latent dimensions of the standard CVAE on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	87
C.3	Marginal posteriors of the 15 latent dimensions of the performance-regularised CVAE on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	87
C.4	Latent traversals along all 15 dimensions of the standard CVAE.	88

C.5	Latent traversals along all 15 dimensions of the performance-regularised CVAE.	90
C.6	Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and performance-regularised BCE-with-logits variants.	94
C.7	Per-dimension KL divergence on the test set for the standard (left) and regularised (right) BCE-with-logits variant.	95
C.8	Marginal posteriors of the 15 latent dimensions of the standard BCE-with-logits variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	95
C.9	Marginal posteriors of the 15 latent dimensions of the performance-regularised BCE-with-logits variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	96
C.10	Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and performance-regularised Continuous Bernoulli variants.	98
C.11	Per-dimension KL divergence on the test set for the standard (left) and regularised (right) continuous-Bernoulli variant.	99
C.12	Marginal posteriors of the 15 latent dimensions of the standard continuous-Bernoulli variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	99
C.13	Marginal posteriors of the 15 latent dimensions of the performance-regularised continuous-Bernoulli variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.	100
C.14	Torque–speed curves of 100 candidate geometries generated for the remaining 4 industrial conditions of §5.5.3, evaluated through the MLP surrogate. Compliant samples are coloured green, non-compliant samples red; target operating points OP1 and OP3 are marked.	101
C.15	Marginal distributions of the 1000 generated samples for the remaining 6 industrial conditions of §5.5.3, both model variants.	103
D.1	Optimisation diagnostics for Condition 2. Convergence (top) and volume baseline comparison (bottom).	107
D.2	Optimisation diagnostics for Condition 3. Convergence (top) and volume baseline comparison (bottom).	109
D.3	Optimisation diagnostics for Condition 4. Convergence (top) and volume baseline comparison (bottom).	110
D.4	Optimisation diagnostics for Condition 5. Convergence (top) and volume baseline comparison (bottom).	111
D.5	Optimisation diagnostics for Condition 6. Convergence (top) and volume baseline comparison (bottom).	112
D.6	Optimisation diagnostics for Condition 7. Convergence (top) and volume baseline comparison (bottom).	113
D.7	Optimisation diagnostics for Condition 8. Convergence (top) and volume baseline comparison (bottom).	114

D.8	FEM validation of the optimisation winners for Conditions 2–5: MLP-predicted vs FEM ground-truth torque–speed curves, with OP1 and OP3 marked. Standard (left) and Regularised (right).	115
D.9	FEM validation of the optimisation winners for Conditions 6–8.	116
D.10	One-pole slice of the winning geometries for Conditions 2–8. Standard (left) and Regularised (right) CVAE.	116

List of Tables

1.1	Geometric parameters defining the double-V rotor topology	5
2.1	Deep generative families compared against the four criteria of §2.3.	18
3.1	Statistical summary of the dataset after physical feasibility filtering ($M = 85\,384$ samples). Geometry-parameter symbols are those of Table 1.1.	30
4.1	Test-set accuracy of the surrogate on the 80-point denormalised torque curve.	40
5.1	Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for both CVAE variants.	48
5.2	The 8 industrial conditions used for the case study, drawn from outside the dataset.	51
5.3	Compliance and range fraction per industrial condition (Conditions 1–8 of Table 5.2), evaluated through the MLP surrogate on 1000 samples per condition.	51
6.1	Hyperparameters of the latent-space optimiser.	60
6.2	Latent-space optimisation winners across the 8 industrial conditions of Table 5.2, for both CVAE variants.	60
A.1	First-order stochastic optimisers commonly used to train MLPs.	76
B.1	Five representative geometry vectors $\mathbf{G}$ from the working dataset, split across two row-blocks for readability.	81
B.2	Condition vectors $\mathbf{C}$ corresponding to the five geometry rows in Table B.1.	82
C.1	Per-feature reconstruction error on the test set, for both CVAE variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent.	85
C.2	Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for the standard and performance-regularised BCE-with-logits variants.	93

C.3	Per-feature reconstruction error on the test set for the standard and performance-regularised BCE-with-logits variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent.	93
C.4	Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for the standard and performance-regularised continuous-Bernoulli variants.	97
C.5	Per-feature reconstruction error on the test set for the standard and performance-regularised continuous-Bernoulli variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent. .	97
C.6	Per-feature range fraction [%] of the 1000 generated candidates per industrial condition (Conditions 1–8 of Table 5.2), for both CVAE variants (Std = standard, Reg = regularised). Averages over the 8 conditions are reported in the last two rows. Parameters DSO-KPQ1. .	102

List of abbreviations and symbols

Abbreviations

AC	Alternating Current
ANN	Artificial Neural Network
BCE	Binary Cross-Entropy
CB	Continuous Bernoulli
CVAE	Conditional Variational Autoencoder
DC	Direct Current
DCP	Design Computation Platform
DGM	Deep Generative Models
DNN	Deep Neural Network
EDA	Exploratory Data Analysis
ELBO	Evidence Lower Bound
EMF	Electromotive Force
EV	Electric Vehicle
FE	Finite Element
FEM	Finite Element Method
GAN	Generative Adversarial Network
IPMSM	Interior Permanent Magnet Synchronous Machine
KL	Kullback–Leibler (divergence)
KPI	Key Performance Indicator
LHS	Latin Hypercube Sampling
LN	Layer Normalisation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
MTPA	Maximum Torque Per Ampere
MTPV	Maximum Torque Per Volt

LIST OF ABBREVIATIONS AND SYMBOLS

NVH	Noise–Vibration–Harshness
OP	Operating Point
OP1, OP3	Operating Point at base / top speed
PCA	Principal Component Analysis
PMSM	Permanent Magnet Synchronous Machine
PP	Performance Profile
ReLU	Rectified Linear Unit
RMS	Root Mean Square
RMSE	Root Mean Square Error
RPM	Revolutions Per Minute
SGD	Stochastic Gradient Descent
SiLU	Sigmoid Linear Unit
SPMSM	Surface Permanent Magnet Synchronous Machine
SVPWM	Space-Vector Pulse-Width Modulation
VAE	Variational Autoencoder

Symbols

Electromagnetic and electrical quantities

B	Magnetic flux density
H	Magnetic field intensity
μ	Magnetic permeability
ϕ	Magnetic flux
ϕ_d, ϕ_q	Flux-map components in the dq plane
λ	Flux linkage of the winding
λ_d, λ_q	dq-frame flux linkages
λ_m	Permanent-magnet flux linkage
I	Current
J	Current density
i_d, i_q	Direct- / quadrature-axis stator currents
u_d, u_q	Direct- / quadrature-axis stator voltages
L_d, L_q	Direct- / quadrature-axis inductances
R_s	Stator phase resistance
$I_{\max}$	Maximum RMS phase-current limit ($I_{inv, \max, RMS}$)
$U_{\max}$	Maximum RMS phase-voltage limit ($U_{ph, RMS, \max}$); SVPWM-derived from $U_{DC, nom}$
$M_{\max}$	Maximum modulation index
$U_{DC, nom}$	Nominal DC-bus voltage
T	Torque
$\hat{T}(\omega)$	Surrogate-predicted torque–speed curve
$\mathbf{y}, \hat{\mathbf{y}}$	Reference / surrogate-predicted torque–speed curve (80-point vector form of $\hat{T}(\omega)$)
$\mathbf{y}_{PCA}, \hat{\mathbf{y}}_{PCA}$	PCA-reduced reference / predicted curve ($\in \mathbb{R}^{d_{PCA}}$)
P	Power
ω	Rotational (mechanical) speed
ω_r	Electrical rotor speed
θ	Rotor angular position

Machine geometry, winding and dataset

$G, \mathbf{G}$	Geometry vector (17 free parameters)
$C, \mathbf{C}$	Condition vector
$\mathcal{G}(\mathbf{C})$	Set of geometries that satisfy condition $\mathbf{C}$
N	Winding ratio
N_{pp}	Number of pole pairs
N_p	Number of parallel winding paths
z_Q	Number of turns per slot
S_{slot}	Copper cross-section per slot

J_{slot}	Slot current density
T_{OP1}, T_{OP3}	Torque at operating points OP1, OP3
$\omega_{OP1}, \omega_{OP3}$	Speed at operating points OP1, OP3
M	Number of dataset samples
ρ_S	Spearman rank-correlation coefficient

Generative model and latent-space optimisation

$\mathbf{x}, \hat{\mathbf{x}}$	Min-max normalised geometry $\mathbf{G} \in [0, 1]^{17}$ (CVAE target) and decoder output
$\mathbf{z}$	Latent code
$\mathbf{z}_0$	Initial (compliant) latent code
d_x, d_z, d_{PCA}	Geometry, latent and PCA-output dimensions
f_θ	Decoder network
q_ϕ	Encoder network (approximate posterior)
p_θ	Decoder likelihood $p_\theta(\mathbf{x} \mathbf{z})$
$\boldsymbol{\mu}, \boldsymbol{\sigma}$	Encoder mean / standard-deviation outputs
$\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x$	Per-feature z -score statistics of the surrogate input
$\boldsymbol{\varepsilon}$	Reparameterisation noise, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$	Standard Normal prior over $\mathbf{z}$
$D_{KL}(\cdot \ \cdot)$	Kullback–Leibler divergence operator
$\mathcal{M}_{CT}$	Speed grid in the constant-torque region (CVAE perf. penalty)
$\mathcal{L}$	Training / optimisation loss
$\mathcal{L}_{recon}, \mathcal{L}_{perf}$	Reconstruction / performance loss terms
δ	Huber-loss threshold
α	Trust-region weight (latent-space optimisation)
β	KL-divergence weight (CH5); performance-violation weight (CH6)
γ	Structural-violation weight (latent-space optimisation)
τ	Free-bits floor on the KL contribution per latent dimension
λ_{perf}	Performance-regularisation weight
η	Learning rate
N_{starts}, n_{iter}	Number of multi-start runs / iterations per run
$v(\hat{\mathbf{x}}), V_{ref}$	Decoded-geometry volume and dataset-mean reference volume
$\sigma(\cdot)$	Logistic sigmoid
$\mathbb{E}$	Expectation
$\mathbb{R}^d$	d -dimensional Euclidean space
$\ \cdot\ $	Euclidean norm

Chapter 1

Introduction

This work examines the design of a double-V interior permanent magnet synchronous machine (IPMSM) for the company Punch Powertrain. More specifically, it investigates the possibility to drastically accelerate the design process by introducing a framework based on deep generative modeling that combines a conditional variational autoencoder for compliant geometry sampling with a subsequent latent-space optimisation stage for refining a candidate towards a chosen secondary objective, while honouring additional constraints built into the optimisation.

This work is the third in a series of Master’s Theses around this subject at Punch Powertrain, building on the work of Senne Lauwers [27] and Stan De Vos [8].

1.1 Industrial Context: Electric Motor Development at Punch Powertrain

Hybrid and electric vehicles are increasingly replacing conventional cars on the road, offering a cleaner and more efficient alternative to internal combustion engines [4, 28]. Driven by stricter emissions regulations and the push to reduce fossil fuel use, car manufacturers are shifting from combustion-based powertrains to electrified ones [4]. The electric traction motor is one of the key components enabling this transition, and demand for these motors is rapidly growing, as they offer high power density, a wide speed range, and high efficiency [7, 28]. Punch Powertrain develops such traction motors for vehicle manufacturers active in this market.

Several motor types are used in electric vehicles, but the interior permanent magnet synchronous machine (IPMSM) has become the dominant choice in modern designs [7, 28]. Within the IPMSM family, multiple rotor configurations exist, several of which are used in commercial traction applications [48]. This work limits itself to the double-V design of the IPMSM.

Within this category, an enormous design space remains, spanned by the internal and external dimensions (e.g. diameter, airgap, magnet size), the orientation and position of the magnets, the type of winding in the stator, and the magnet and lamination materials.

When designing a machine for a vehicle manufacturer, Punch Powertrain receives client requirements such as the desired torque and power output, electrical limits, and thermal limits, and tries to find particular designs within that design space that are compliant with those requirements while minimising cost, size, and mass and maximising efficiency. Currently, the company approaches this design process with an in-house tool called the ZONDA Toolchain (see §1.1.2).

1.1.1 Optimization Objectives

The design of an electric machine for automotive traction is fundamentally a constrained multi-objective optimisation problem. The goal is to minimise manufacturing cost, mass, and physical size while maximising energy-conversion efficiency. These objectives are inherently conflicting: reducing material usage lowers cost and mass but may compromise torque density or thermal margins, while increasing efficiency often requires additional active material or tighter tolerances that raise manufacturing expense. The optimiser therefore navigates this trade-off space, typically returning a Pareto front of non-dominated solutions from which engineers select based on secondary criteria or client preference. The formal multi-objective statement is given in Appendix A.1.

Constraints fall into performance, electrical, and thermal categories, discussed hereafter.

Performance constraints require that the motor achieve at least the torque and power specified by the client’s performance profile at every relevant operating point. The performance profile defines four relationships: peak and continuous torque versus speed, and peak and continuous power versus speed (Figure 1.1). The peak curves define the maximum instantaneous output the machine must deliver as a function of speed; the duration over which peak torque can be applied is bounded by the thermal limits, so it is often only available during brief periods (fast accelerations). The continuous curves represent the sustained output that can be maintained indefinitely without exceeding thermal limits, which requires a full thermal analysis to verify that winding, magnet, and lamination temperatures remain within material-specific bounds under prolonged operation.

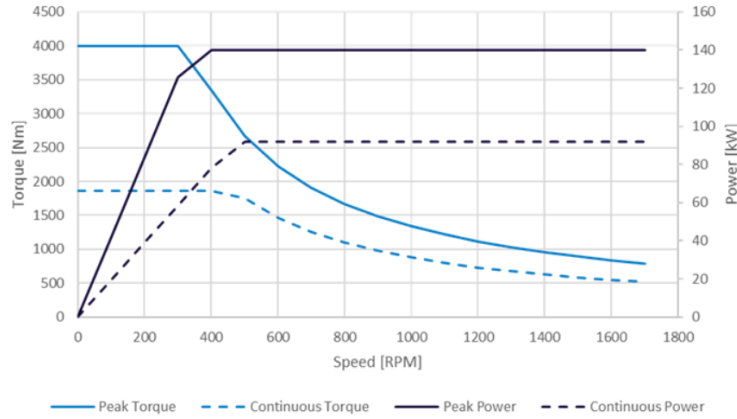


FIGURE 1.1: Vehicle-level torque–speed and power–speed requirement characteristics, as typically specified by the client at the wheels, showing peak and continuous operating envelopes. Source: Punch Powertrain

The characteristics in Figure 1.1 are specified at *vehicle level*, i.e. at the wheels, which is the form in which a client typically delivers the requirement. Punch Powertrain translates this vehicle-level requirement into an equivalent *motor-level* performance profile through the selected (optimal) gear ratio. From this point onward, all performance quantities in this thesis refer to the motor level; the torque magnitudes encountered in later chapters are therefore roughly an order of magnitude smaller than the wheel-level values shown here.

Electrical constraints arise from the vehicle’s power electronics and battery system. The inverter sets the maximum phase current $I_{\max}$ that can be delivered to the motor windings, and the battery pack sets the DC-bus voltage $U_{\max}$ available to the inverter. Together these two limits bound the achievable torque–speed envelope.

Thermal constraints ensure that no component (windings, magnets, or lamination stack) exceeds its material-specific temperature limit during sustained operation. Peak operating points may be thermally permissible for short durations, but continuous operating points must be sustainable indefinitely without thermal runaway. The thermal module of the Design Computation Platform evaluates this by computing the time required for each component to reach its limit under representative duty cycles.

Beyond these primary categories, the geometry must also be mechanically viable. Centrifugal forces at high rotational speeds induce stresses on the rotor laminations and magnet bridges; designs that would fail structurally at peak speed are rejected. Additional secondary criteria such as demagnetisation margin, cogging torque amplitude, and noise–vibration–harshness (NVH) may further constrain the feasible region depending on the application.

1.1.2 The ZONDA Toolchain

The ZONDA Toolchain is the in-house framework that Punch Powertrain currently uses to translate a set of client requirements into a finalised motor design. It consists of an optimisation loop, shown in Figure 1.2, in which three components are chained together: a parametric description of the machine, a computation module that evaluates this description, and an optimisation algorithm that drives the loop.

At each iteration, a candidate design is represented as a vector of geometry parameters together with a choice of materials. These are passed to the Design Computation Platform (DCP), which serves as the evaluation engine of the toolchain and combines analytical models with finite element method (FEM) simulations to compute the Key Performance Indicators (KPIs) that describe the candidate. The KPIs are returned to the optimisation algorithm, which compares them against the objectives and constraints, proposes a new geometry, and iterates until a configuration is found that satisfies all constraints while optimising the objectives of §1.1.1.

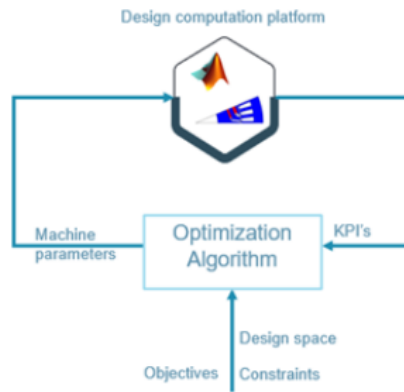


FIGURE 1.2: Closed-loop optimisation between machine parameters and KPI evaluation via the Design Computation Platform. Source: Punch Powertrain

Machine Parametrization

The ZONDA toolchain begins every design iteration from a compact numerical description of the motor geometry. Because the double-V IPMSM cross-section is periodic under the pole count (assuming nominal conditions), the full machine is reduced by symmetry to a single pole, and the parametrization is attached to that slice rather than to the complete geometry. This keeps the input dimensionality small without losing any information. Two practical examples of such geometry slices are shown in Figure 1.3.

The parametrization adopted in this thesis is listed in Table 1.1 and consists of 19 scalar variables. Two of them, KDV and KSH, are held at fixed values (0 and 0.7 respectively). The remaining 17 span continuous ranges and form the effective design space. Each variable corresponds to a specific geometric feature of the pole slice, such as the stator outer diameter DSO (expressed in metres in Table 1.1) and the mechanical air gap GAP (in millimetres). The other variables are dimensionless ratios that fix proportions of the pole slice: KAR (Key Aspect Ratio), for instance, is the ratio of stack length to DSO, and the remaining ratios analogously parametrise magnet layer widths, radial proportions, and tooth geometry. The intervals in Table 1.1 bound the region of the design space in which the downstream DCP calculations are physically meaningful.

For the purposes of this work, the lamination and magnet materials are kept constant at LAMMAT = HILITE N020 MEAS and MAGMAT = N42UH. Material substitution is therefore not treated as a design variable.

TABLE 1.1: Geometric parameters defining the double-V rotor topology

Parametrization			
Variable	Range	Variable	Range
DSO	0.18 - 0.25	KPQ1	0.1 - 0.9
GAP	0.5 - 1.2	KPQ2	0.1 - 0.9
KAR	0.4 - 1.1	KQI	0.5 - 0.8
KBETA	0.2 - 0.35	KQIVAR	0.3 - 0.55
KDV	0	KR0	0.55 - 0.7
KFD	1.2 - 2.6	KR1	0.3 - 0.7
KGAMMA	0.05 - 0.3	KSH	0.7
KMQ1	0.4 - 0.9	KSR	0.65 - 0.8
KMQ2	0.4 - 0.9	KTW	0.4 - 0.6
KYH	0.4 - 0.6		

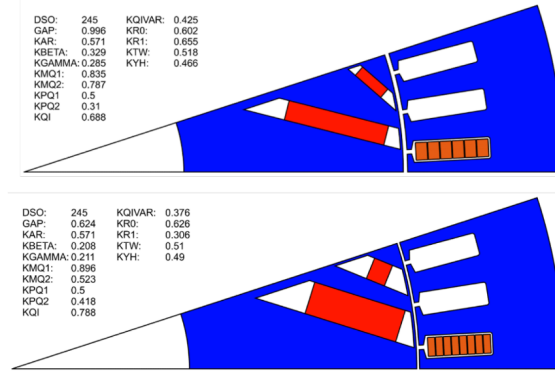


FIGURE 1.3: Two valid double-V IPMSM rotor designs and their corresponding parametrizations.

Design Computation Platform

The DCP is the evaluation engine of the ZONDA Toolchain. For a candidate described by a geometry vector $\mathbf{G}$ and a material selection, it returns the six KPIs on which the optimiser acts (Figure 1.4): structural integrity, cost, performance, efficiency, thermal behaviour, and NVH. These KPIs constitute the inputs to the objective functions and constraints of §1.1.1; the per-module description and the internal sequencing of the three FEM-dependent modules (performance, efficiency, thermal) are documented in Appendix A.2.

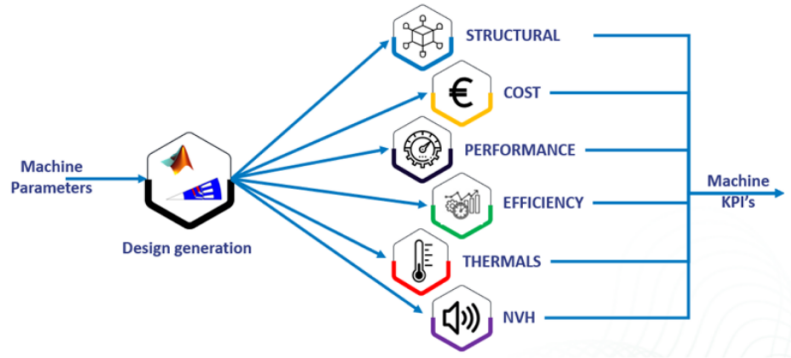


FIGURE 1.4: KPI calculation engine.

Computational Considerations

The cost of the ZONDA toolchain is concentrated in the DCP’s KPI evaluation. Three of its six modules (performance, thermal behaviour, and efficiency) rely on 2D finite element simulations of the parametrised geometry; the analytical modules (structural, cost, NVH) are negligible in comparison. A full KPI evaluation of a single candidate geometry consequently requires on the order of 20 minutes of single-core CPU time on modern hardware [27, 8].

The large search space makes the optimisation as a whole especially costly. The optimiser operates over the full 19-parameter design space, and for every candidate the FEM step must verify operational compliance across the torque–speed envelope under the client’s electrical limits. Because the evolutionary loop issues thousands of such candidates before converging, the aggregate workload remains on the order of several days to roughly one week even when distributed across a 48-core node [27]. The combination of a broad design space and a FEM-based feasibility check per iteration turns each customer case into an exhaustive search.

1.2 Problem Statement and Research Objectives

Problem Statement

The ZONDA Toolchain of §1.1.2 solves the forward design problem: given a geometry $\mathbf{G}$, evaluate its KPIs through the DCP. The inverse direction (given a set of client requirements, identify geometries that satisfy them) is handled by wrapping this forward evaluator in the optimisation loop of Figure 1.2, which searches the design space by issuing thousands of forward queries. This is exactly the computational bottleneck identified earlier: the design optimisation problem is solved through repeated exhaustive forward evaluation for each candidate geometry in the entire design space.

The previous theses in this series [27, 8] reduced the cost of a single forward evaluation by surrogating the flux-map FEM step. This work takes a different route: it tackles the inverse problem directly (i.e. given client requirements, generate motor geometries that satisfy them). The investigation is scoped to performance compliance under electrical limits; the evaluation of thermal, efficiency, NVH, and cost KPIs lies outside this work.

The client’s requirements form a requirement vector $\mathbf{C}$, and the design task is to find geometries $\mathbf{G}$ that satisfy it. This inverse problem has no unique answer: many distinct geometries (the family $\mathcal{G}(\mathbf{C})$) satisfy the same requirements. A deterministic regressor mapping $\mathbf{C}$ to $\mathbf{G}$ trained under a squared-error loss approximates the conditional mean of $\mathbf{G}$ given $\mathbf{C}$, which provides no diversity for downstream selection. Moreover, the average of several valid designs is not necessarily itself a valid design [5], so the predicted geometry is not guaranteed to be feasible. The goal is therefore to characterise $\mathcal{G}(\mathbf{C})$ and generate many compliant candidates from it. The precise form of $\mathbf{C}$ and the criterion for satisfying it follow in Chapter 3.

Beyond characterising $\mathcal{G}(\mathbf{C})$ itself, a second design question arises naturally: given a set of compliant candidates, can they be further refined towards a chosen secondary objective? This question is addressed in Chapter 6.

Proposed Approach

The proposed framework is structured in two stages. The first stage learns to generate geometries that satisfy a given requirement, and at inference time it is realised as a pipeline combining two models:

1. A **Conditional Variational Autoencoder (CVAE)** conditioned on the requirements $\mathbf{C}$, which samples diverse candidate geometries. This is the generative component that supplies diversity; its selection among other deep generative models is motivated in §2.3.
2. A **frozen performance surrogate** (a multilayer perceptron) that maps a geometry together with the electrical limits to its performance profile. The electrical limits appear explicitly on the input side because the performance profile is not a function of geometry alone (see §2.1.3).

A key property of the CVAE is that it embeds the geometry in a compact latent space in which compliance with $\mathbf{C}$ is already encoded. In the second stage, this latent space is used as the search domain for a gradient-based optimisation, navigating towards a chosen objective while remaining within the compliant region by utilising the frozen surrogate of Chapter 4. In this work, as a proof of concept, the objective is minimal volume, with performance-compliance and structural-integrity constraints built into the optimisation (Chapter 6).

Two CVAE variants are investigated: a baseline trained with the standard variational objective, and a performance-regularised variant in which the frozen surrogate is inserted into the training loss as a differentiable penalty on performance-compliance violation.

Research Objectives

1. Train a surrogate model that predicts a motor’s peak torque-speed envelope from its geometry and electrical constraints (Chapter 4).
2. Develop a standard CVAE that learns to generate motor geometries from client requirements (§5.2, Chapter 5).
3. Extend to a performance-regularised CVAE using the surrogate of (1) as auxiliary training signal (§5.3, Chapter 5).
4. Demonstrate, as a proof of concept, that the learned latent space supports a downstream optimisation stage: refine a sampled candidate by gradient descent towards minimal volume, with performance-compliance and structural-integrity constraints built into the optimisation, using the surrogate of (1) as the in-loop performance evaluator (Chapter 6).

Chapter 2

Theoretical Background

The framework proposed in §1.2 is based on three pillars of theory: the electromagnetic operation of the target machine (§2.1), the working principles of artificial neural networks (§2.2), and the fundamentals of deep generative models (§2.3). Section 2.4 then consolidates the network-design and training choices common to the surrogate and the CVAE, and Section 2.5 closes the chapter with a review of prior work in AI-based electric-machine design.

A four-pole instance of the double-V IPMSM topology used throughout this work is shown in Figure 2.1.

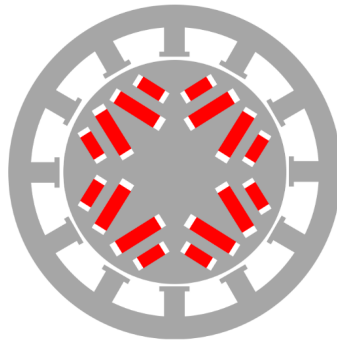


FIGURE 2.1: Double V-shaped IPMSM design with 4 poles. Source: Punch Powertrain

2.1 The Interior Permanent Magnet Synchronous Machine

2.1.1 Classification of PMSMs

Modern automotive traction relies on AC machines, with both induction and synchronous variants present in commercial powertrains [28, 48]. Permanent magnet synchronous machines (PMSMs) replace the rotor field winding with rare-earth mag-

nets, raising the air-gap flux density and avoiding rotor conduction losses, which yields a higher torque and power density at higher efficiency [39, 28]. Within the PMSM family, the interior PMSM embeds the magnets inside the rotor laminations: this protects them mechanically at high speed and introduces magnetic saliency ($L_d \neq L_q$), which adds a reluctance-torque component that underlies the wide field-weakening range required by automotive duty cycles [48]. The IPMSM is consequently the dominant traction topology and the one adopted here. Within the IPMSM family several rotor arrangements are used in commercial designs (single-V, double-V, delta, U-shape, illustrated in Figure 2.2); this work focuses on the double-V configuration.

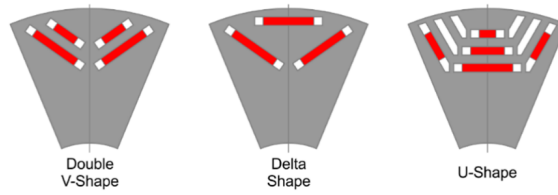


FIGURE 2.2: Three common rotor topologies in the IPMSM family.

2.1.2 Electromagnetic Model and Working Principles

This subsection introduces the minimal electromagnetic background needed to describe how an IPMSM converts electrical input into mechanical torque, and how its torque–speed envelope (the performance profile) emerges from physical limits. Only the relations actually used in the rest of this thesis are retained.

Flux Linkage and Induced EMF

The magnetic flux ϕ that passes through a surface S is defined as

$$\phi = \int_S \mathbf{B} \cdot d\mathbf{A} \quad [\text{Wb}], \quad (2.1)$$

where $\mathbf{B}$ is the magnetic flux density. For a coil of N turns, the total flux linkage is

$$\lambda = N\phi \quad [\text{Wb}], \quad (2.2)$$

and Faraday’s law gives the voltage induced in the coil:

$$e = -\frac{d\lambda}{dt}. \quad (2.3)$$

The flux density and the magnetic field intensity $\mathbf{H}$ are related by the constitutive relation $\mathbf{B} = \mu\mathbf{H}$, where μ is the magnetic permeability of the material. In the iron laminations of the stator and rotor, μ is strongly nonlinear: at high field intensities the iron saturates and μ drops sharply, making the flux a nonlinear function of the applied current. This nonlinearity prevents closed-form analytical evaluation of the flux maps and is the primary reason why FEM is used to compute them in the ZONDA toolchain (see §2.1.3).

The Arbitrary Reference Frame

The three-phase stator quantities (a, b, c) are defined on fixed axes spaced 120° apart, in which the inductances vary periodically with rotor position and complicate both analysis and control. To obtain a tractable torque expression, the three-phase quantities are mapped to a rotating orthogonal $(q, d, 0)$ frame using the Clarke and Park transforms [32]; the explicit transform matrix is reproduced in Appendix A.3. When the frame is locked to the rotor by setting $\omega = \omega_r$ and aligning the d -axis with the permanent-magnet flux vector, all inductances become constant and all voltages and currents become DC quantities in steady state. This specific choice is the *synchronous reference frame* used throughout the remainder of this thesis. Figure 2.3 illustrates this frame for the double-V topology.

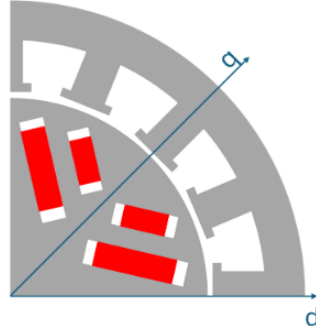


FIGURE 2.3: Synchronous (rotor-locked) reference frame for the double-V IPMSM.

Voltage and Torque Equations in the dq Frame

In the synchronous reference frame, the stator flux linkages are

$$\lambda_d = L_d i_d + \lambda_m, \quad \lambda_q = L_q i_q, \quad (2.4)$$

where L_d and L_q are the direct- and quadrature-axis inductances, λ_m is the flux linkage established by the permanent magnets along the d -axis, and i_d , i_q are the dq stator currents. The steady-state stator voltage equations are

$$u_d = R_s i_d - \omega_r L_q i_q, \quad u_q = R_s i_q + \omega_r (L_d i_d + \lambda_m), \quad (2.5)$$

where R_s is the stator phase resistance and ω_r is the electrical rotor speed. The electromagnetic torque follows from the instantaneous power balance [26]:

$$T = \frac{3}{2} N_{pp} (\lambda_d i_q - \lambda_q i_d) = \frac{3}{2} N_{pp} [\lambda_m i_q + (L_d - L_q) i_d i_q], \quad (2.6)$$

where N_{pp} is the number of pole pairs. The first term, $\lambda_m i_q$, is the *magnet torque*; it depends only on i_q and is analogous to the torque of a separately-excited DC machine. The second term, $(L_d - L_q) i_d i_q$, is the *reluctance torque*; it arises from the inductance asymmetry caused by the embedded magnet geometry.

For an IPMSM, $L_d < L_q$, so the reluctance torque is maximised by driving i_d negative. This is the mechanism that makes the IPMSM both more torque-dense and better suited to field weakening than a surface-mounted machine, where $L_d \approx L_q$ and reluctance torque vanishes.

Operating Envelope and the Performance Profile

The torque that a machine can deliver at a given speed is bounded by two hardware limits. Let $\hat{I}_{\max} = \sqrt{2} I_{\max}$ and $\hat{U}_{\max} = \sqrt{2} U_{\max}$ denote the peak-amplitude counterparts of the RMS limits defined in (3.3). The first hardware bound is the *current limit* of the inverter:

$$i_d^2 + i_q^2 \leq \hat{I}_{\max}^2, \quad (2.7)$$

which traces a circle of radius $\hat{I}_{\max}$ in the dq current plane. The second is the *voltage limit* imposed by the DC-bus:

$$u_d^2 + u_q^2 \leq \hat{U}_{\max}^2. \quad (2.8)$$

Substituting the steady-state voltage equations (2.5) and neglecting stator resistance, the voltage constraint becomes an ellipse centred at $(-\lambda_m/L_d, 0)$ in the dq plane:

$$\left(\frac{i_d + \lambda_m/L_d}{\hat{U}_{\max}/(\omega_r L_d)} \right)^2 + \left(\frac{i_q}{\hat{U}_{\max}/(\omega_r L_q)} \right)^2 \leq 1. \quad (2.9)$$

The interaction of these two constraints determines three distinct operating regimes, summarised by the trajectory A–B–C–D in Figure 2.4 [34, 43]. Below the base speed (Mode I, MTPA), the voltage ellipse is wider than the current circle and the optimal strategy minimises copper losses by placing the dq current vector on the Maximum-Torque-Per-Ampere locus; the maximum torque is set by the intersection of MTPA and the current circle (point A). Above the base speed (Mode II, flux weakening), the ellipse contracts and both constraints are simultaneously active: the operating point slides along the current circle towards more negative i_d (segment A–B), weakening the air-gap flux to push back the voltage limit at the cost of reduced torque. At still higher speeds (Mode III, MTPV), only the voltage constraint remains active and the operating point follows the ellipse towards its centre (segment B–C). Point C marks the maximum mechanical speed, where rotor stresses become binding; point D, at the centre of the voltage ellipse, is the theoretical asymptote approached only at infinite speed.

Translating from the dq current plane to the speed domain, the combination of these three regimes yields the performance profile (PP): the maximum torque envelope $T_{\max}(\omega)$ across the full speed range $[0, \omega_{\max}]$, with the corresponding power envelope $P_{\max}(\omega) = T_{\max}(\omega) \cdot \omega$. This envelope is illustrated in §1.1.1 (Figure 1.1). How the performance profile is abstracted to the discrete operating points OP1 and OP3 used as compliance criteria in this thesis is treated in §3.2.

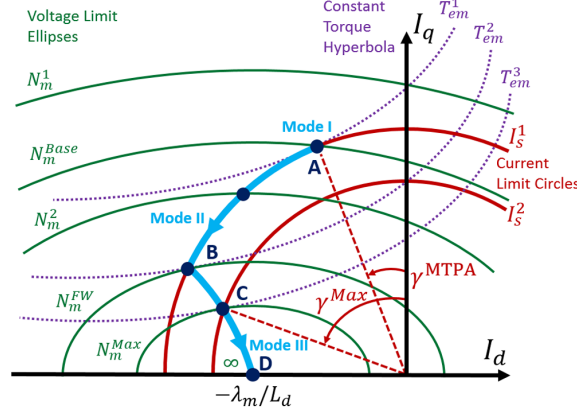


FIGURE 2.4: Operating modes of an IPMSM in the dq current plane, along the maximum-torque trajectory A–B–C–D. Reproduced from [34].

2.1.3 From Geometry to Performance Profile

The DCP translates a geometry $\mathbf{G}$ into a performance profile in three sequential steps (see Figure A.1). The full workflow, including the inputs used at each stage, is reproduced in Figure 2.5. This subsection establishes the physical content of each step, in the order in which the ZONDA toolchain executes them. The data-generation pipeline used in this thesis is a modified version of this workflow and is treated separately in §3.1.

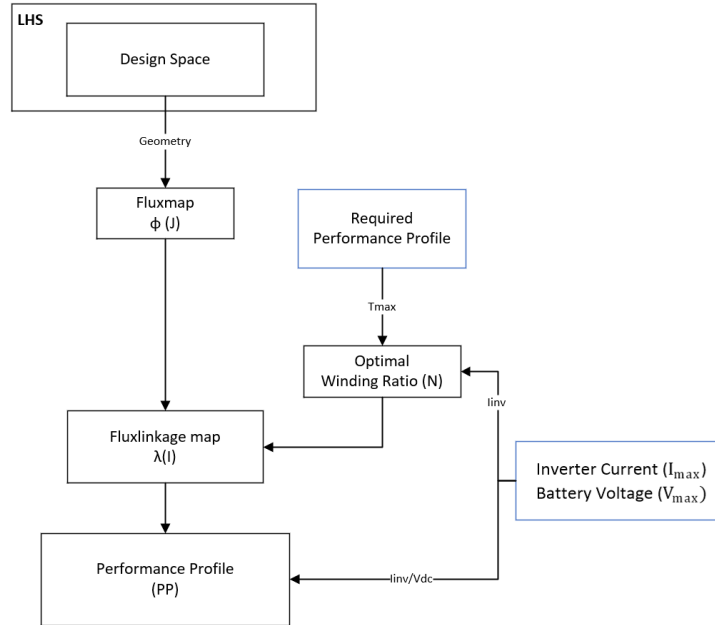
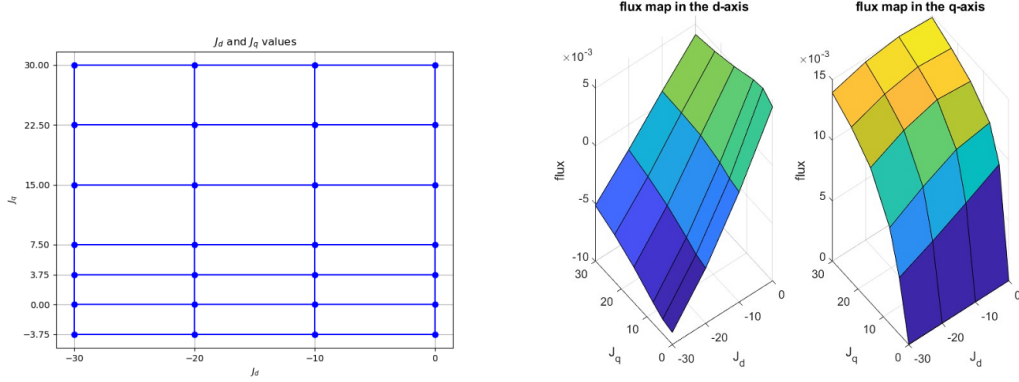


FIGURE 2.5: Conventional data generation workflow, used in the ZONDA Toolchain.

Step 1: Flux-Map Generation

The first step computes the *flux map* of the candidate machine: the pair of functions $\phi_d(J_d, J_q)$ and $\phi_q(J_d, J_q)$, evaluated on a discrete grid in the dq current-density plane for a fixed pole-slice geometry and material set. The flux map is the lowest-level electromagnetic descriptor of the design, from which all downstream torque and voltage quantities follow.

Because the constitutive relation $\mathbf{B} = \mu\mathbf{H}$ in the iron laminations is strongly non-linear due to magnetic saturation (§2.1.2), no closed-form expression for ϕ_d and ϕ_q exists. The ZONDA toolchain consequently solves the field problem numerically, with 2D finite element analysis on the 24-point (J_d, J_q) grid of Figure 2.6a; a representative output for a double-V geometry is shown in Figure 2.6b. A neural-network surrogate that bypasses this FEM step has been developed in prior work [8] and is the entry point of the data-generation pipeline used in this thesis (§3.1).



(A) 24-point (J_d, J_q) grid over which the d- and q-axis flux maps are evaluated.

(B) Representative d- and q-axis flux maps obtained from 2D FEM simulation for a fixed double-V IPMSM geometry.

FIGURE 2.6: Flux map generation and resulting d- and q-axis flux maps.

The flux map alone does not yet yield torque: the dq torque expression (2.6) is written in terms of flux *linkage* λ , not flux ϕ , and the conversion between the two depends on how the stator coils are wound.

Step 2: From Flux Map to Flux-Linkage Map: the Winding Ratio

The flux map is set by the rotor and stator geometry, the magnet and lamination materials, and the imposed slot current density, but it is independent of how the stator coils are connected externally. The flux *linkage* λ , in contrast, depends on the number of turns the winding presents to the same magnetic flux. The two are bridged by the *winding ratio* N , defined as the ratio of the number of turns per coil

z_Q to the number of parallel paths N_P :

$$N = \frac{z_Q}{N_P}, \quad \lambda_d = N \phi_d, \quad \lambda_q = N \phi_q. \quad (2.10)$$

When altering this N , the FE field solution (and hence the flux per slot) at fixed J is unchanged. But the corresponding physical phase current I still depends on N . Following [44], the rated phase current corresponding to a slot current density J in a slot of cross-section S_{slot} is

$$I = \sqrt{I_d^2 + I_q^2} = \frac{\sqrt{2}}{2} J S_{slot} \frac{N_P}{z_Q} = \frac{\sqrt{2}}{2} J S_{slot} \frac{1}{N}, \quad (2.11)$$

Equations (2.10) and (2.11) together describe the two central scalings of rewinding: increasing N scales the flux linkage λ linearly while scaling the physical phase current I inversely. A single FE flux map therefore yields the flux-linkage map for every N ; the underlying Poisson invariance is reproduced in Appendix A.4.

The performance profile, in contrast, does depend on N , and this is why N must be tracked as a separate design parameter alongside the geometry. The argument splits naturally between the two regimes of the operating envelope (§2.1.2):

- *Constant-torque (current-limited) region.* The dq torque expression (2.6) contains the product λI . Since $\lambda \propto N$ and $I \propto 1/N$, this product is invariant: at a fixed inverter current limit $I_{\max}$, the maximum achievable torque $T_{\max}$ is the same for every N .
- *Field-weakening (voltage-limited) region.* The induced voltage scales as $u \propto \omega \lambda \propto \omega N$. At a fixed inverter voltage limit $U_{\max}$, the voltage constraint (2.8) becomes binding at a base speed $\omega_b \propto U_{\max}/N$, so a higher N pushes the onset of field weakening to a lower speed and shifts the entire $T_{\max}(\omega)$ curve above ω_b accordingly.

A single $(\mathbf{G}, \text{materials})$ pair therefore yields a distinct PP per choice of N , so N must enter the condition vector $\mathbf{C}$ alongside the geometry and electrical limits (see §3.2).¹

Step 3: From Flux-Linkage Map to Performance Profile

With the flux-linkage map (λ_d, λ_q) fixed by Steps 1 and 2, the performance profile is recovered by applying the operating-envelope construction of §2.1.2 numerically. The toolchain sweeps speed from 0 to $\omega_{\max}$ in fixed increments; at each ω , the MTPA locus, the current-limit circle and the speed-dependent voltage ellipse are intersected on the dq plane, with torque evaluated through the discrete flux-linkage map by interpolation, and the maximum feasible torque $T_{\max}(\omega)$ is read off. The output of the three-step chain is one PP curve per tuple $(\mathbf{G}, U_{\max}, I_{\max}, N)$.

¹In the target machine N is not continuous: it is restricted to the discrete set $\{1, 2, 3, 4\}$ by the hairpin winding technology used. See Appendix A.4 for the technological background.

2.2 Artificial Neural Networks

The data-driven framework proposed in this thesis rests on artificial neural networks (ANNs) as the workhorse for both a regression surrogate (Chapter 4) and the parameterised conditional distribution at the core of the generative model (Chapter 5). Both are instances of the multilayer perceptron (MLP) family: a feedforward MLP that maps a geometry-condition vector to a performance profile, and a pair of MLPs forming the encoder and decoder of a conditional variational autoencoder.

A feedforward MLP is a stack of affine-nonlinear layers. Writing $\mathbf{h}^{(0)} = \mathbf{x}$ for the input, each subsequent layer evaluates

$$\mathbf{h}^{(\ell)} = f\left(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \quad \ell = 1, \dots, L, \quad (2.12)$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the trainable weights and biases of layer ℓ and f is a non-linear activation applied element-wise; the final layer typically omits or replaces f depending on the type of target. A schematic with one hidden layer is shown in Figure 2.7. A single hidden layer of sufficient width is in principle a universal approximator of continuous functions on compact domains [20], but at fixed parameter budget deeper networks tend to generalise better and are the prevailing design choice for the regression and generative tasks considered here [14].

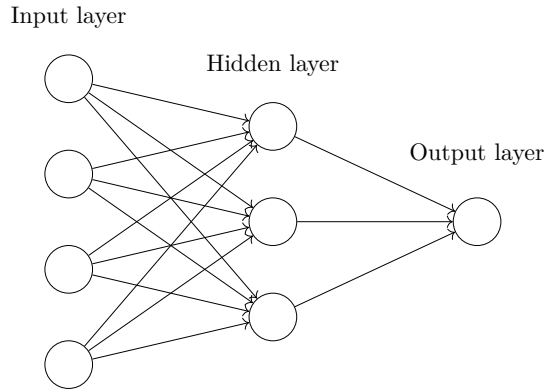


FIGURE 2.7: Schematic of a feedforward multilayer perceptron with one hidden layer.

Training, Validation and Test Data

To assess whether an MLP has learned a transferable mapping rather than memorised its training points, the available samples are partitioned into three disjoint subsets. The training set drives the parameter updates; the validation set is consulted after each pass through the data to monitor generalisation and to select hyperparameters; the test set is held out until all design decisions have been frozen, so that its loss reflects performance on genuinely unseen inputs.

Training proceeds iteratively. In each step the network performs a forward pass on a mini-batch of inputs to obtain predictions, a scalar loss is evaluated against the targets, gradients of the loss with respect to every weight and bias are computed by backpropagation [41], and an optimiser updates the parameters using these gradients. One full traversal of the training set constitutes an epoch; mini-batching the data within an epoch trades exact gradient direction for stochastic gradient steps that fit in memory and act as an implicit regulariser. A persistent gap, in which the training loss continues to decline while the validation loss flattens or rises, signals overfitting. The numerical sizes of the splits used in this work are deferred to Chapter 3 since they are dataset-specific.

A standard textbook treatment of these mechanics can be found in [14].

2.3 Deep Generative Models

Section 1.2 formulated the inverse design problem as the task of learning the conditional distribution $p(\mathbf{G} \mid \mathbf{C})$ rather than its conditional mean. A standard regressor would collapse the one-to-many structure of the inverse map and may even return geometries that are themselves infeasible [5]. The rest of this chapter introduces the deep generative models suited to this task and motivates the choice made for this thesis. Three criteria, derived from the role this model has to play, guide the model selection in §2.3.1: (i) conditioning on a multi-variate requirement vector $\mathbf{C}$ rather than a categorical class label; (ii) sample diversity within the feasible set $\mathcal{G}(\mathbf{C})$; (iii) sub-second inference per candidate. A fourth, downstream criterion follows from the performance regulariser of Chapter 5: the decoder must be differentiable so that gradients of an external surrogate-based penalty can be propagated through it.

2.3.1 Families of Deep Generative Models

Five families dominate the modern deep generative literature [42, 46]: autoregressive models, normalizing flows, diffusion models, generative adversarial networks (GANs), and variational autoencoders (VAEs). Each has demonstrated success in problems with similar structure, and several would be defensible choices here. Table 2.1 consolidates their training signal, conditioning mechanism, inference cost, decoder differentiability, and precedents in adjacent inverse-design problems.

Autoregressive models. Autoregressive models factorise the joint distribution into a product of conditionals, $p(\mathbf{x}) = \prod_i p(x_i \mid x_{<i})$, and are highly effective on data with a natural ordering. The 17-dimensional geometry vector of Table 1.1 has no such ordering, which makes this family less natural here.

Normalizing flows. Normalizing flows form a chain of invertible mappings that turn a simple base distribution into the target, yielding an exact likelihood. The invertibility constraint restricts the architecture to layers with cheaply computable Jacobian determinants.

Diffusion models. Diffusion models achieve state-of-the-art sample quality and have been applied to conditional synthesis in adjacent domains [13]. Their iterative denoising procedure requires tens to hundreds of network evaluations per sample, which makes them more expensive than single-step alternatives when the pipeline must produce many candidates per requirement vector.

Generative adversarial networks (GANs). GANs train a generator and a discriminator adversarially and tend to produce sharp samples, but lack an explicit likelihood and are susceptible to mode collapse, which competes with the diversity criterion. Conditional variants compensate with auxiliary losses, as the CDC-GAN of Akbari and Lowther [1] demonstrates for axial-flux machines.

Variational autoencoders (VAEs). VAEs train through an explicit variational lower bound on the log-likelihood and use a structured Gaussian latent. Conditioning on $\mathbf{C}$ proceeds by concatenation at the encoder and decoder inputs, and the decoder is a plain MLP. Precedents include Barmada et al. for electromagnetic field reconstruction [3], Parekh et al. for multi-objective optimisation of electrical machines [37], and inverse-design applications outside engineering [12, 47].

TABLE 2.1: Deep generative families compared against the four criteria of §2.3.

Family	Training signal	Conditioning on $\mathbf{C}$	Inference per sample	Diff. de-coder	Precedent
Autoregressive	Exact log-likelihood (factorised)	Direct	Sequential	N/A	—
Normalizing flow	Exact log-likelihood (change-of-variables)	Architecturally constrained	Single pass	Yes	—
Diffusion	Score / denoising loss	Direct	Iterative (~ 10 – 100 steps)	Yes	[13]
GAN / cGAN	Adversarial (no explicit likelihood)	Direct	Single pass	Yes	[1]
VAE / CVAE	ELBO (lower bound on log-likelihood)	Direct	Single pass	Yes	[3, 37]

Model selection. The CVAE is adopted in this work. The choice is a pragmatic trade-off rather than a claim of universal superiority: a conditional GAN such as CDC-GAN [1] is a defensible alternative, and a diffusion model would likely produce sharper samples at higher cost. Three properties of the CVAE nonetheless align particularly well with the rest of this thesis; they are spelled out in section 2.4, once the model itself has been introduced.

2.3.2 The Variational Autoencoder

A VAE couples two networks, much like an ordinary autoencoder: an *encoder* that compresses a high-dimensional input (here, a motor geometry) into a smaller set of latent numbers, and a *decoder* that reconstructs the input from those numbers [14]. What makes it *variational* is the encoder’s output: instead of a single latent point it produces a probability distribution (a Gaussian) over the latent space, and the decoder is trained to reconstruct the input from samples of that distribution [24]. Training drives these per-input distributions to overlap and to fill a simple, continuous latent space; once trained, new latent points drawn from that space decode into new, plausible geometries, which is what makes the model *generative* rather than merely compressive [25].

The VAE [24] models the data as the marginal of a latent-variable model with a Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ parameterised by a neural network. Because direct maximisation of $\log p_\theta(\mathbf{x})$ is intractable, a second network, the encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$, is introduced as a tractable approximation to the true posterior. The standard variational derivation (reproduced in Appendix A.6) yields the *evidence lower bound* (ELBO):

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \text{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})). \quad (2.13)$$

The first term is a *reconstruction* term; under a Gaussian decoder it reduces (up to constants) to the negative squared error between $\mathbf{x}$ and the decoder output $\hat{\mathbf{x}}_\theta(\mathbf{z})$, recovering a standard MSE reconstruction loss. The second term is a KL *regulariser* that pulls the encoder posterior towards the prior; for diagonal Gaussian posteriors and the standard prior it admits a closed form, differentiable in $(\boldsymbol{\mu}, \boldsymbol{\sigma})$, which is contributed to the loss directly (see Appendix A.6).

Reparameterisation trick. The reconstruction term in (2.13) involves an expectation under q_ϕ , whose parameters are exactly what is being differentiated. Sampling $\mathbf{z} \sim q_\phi$ directly blocks gradient flow back into ϕ . The reparameterisation trick [24] writes

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (2.14)$$

moving the stochasticity into a parameter-free noise variable $\boldsymbol{\varepsilon}$. Gradients with respect to $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ now flow deterministically and back-propagation proceeds through the encoder.

Figure 2.8 contrasts the original and reparameterised computational graphs: in the reparameterised form $\mathbf{z}$ becomes a deterministic function of ϕ , $\mathbf{x}$, and the auxiliary noise ϵ , so gradients $\nabla_{\phi} f$ can be computed by standard back-propagation.

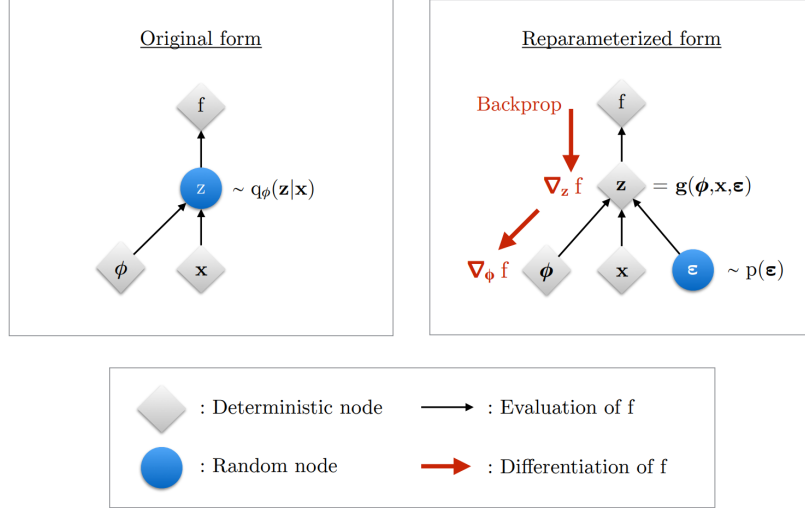


FIGURE 2.8: Illustration of the reparameterisation trick [25]. Original (left) and reparameterised (right) computational graphs for $\mathbf{z} \sim q_{\phi}(\mathbf{z} | \mathbf{x})$.

β -VAE. A useful generalisation [18] attaches a scalar weight β to the KL term:

$$\mathcal{L}_{\beta}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_{\phi}}[\log p_{\theta}(\mathbf{x} | \mathbf{z})] - \beta \cdot \text{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})). \quad (2.15)$$

The coefficient β controls a trade-off between reconstruction fidelity and the regularity of the latent space: $\beta > 1$ increases the pressure towards the prior, whereas $\beta < 1$ relaxes it in favour of accurate reconstructions. The CVAE of Chapter 5 uses $\beta < 1$; the precise value is given there.

Posterior collapse and KL annealing. A failure mode specific to the VAE is *posterior collapse*: the encoder degenerates to $q_{\phi}(\mathbf{z} | \mathbf{x}) \approx p(\mathbf{z})$, the KL term vanishes, and the latent code carries no information about $\mathbf{x}$. Two mitigations are used in this work. *KL annealing* starts training at $\beta = 0$ and ramps the coefficient up linearly: the model first learns to reconstruct using an informative latent, after which the regularisation gradually constrains the posterior [6]. *Free bits* replaces the per-dimension KL with a clamped version that is not penalised below a small threshold, preventing the optimiser from trivially zeroing the latent dimensions one by one [23]. The concrete schedule and threshold used in this thesis are reported in Chapter 5.

2.3.3 The Conditional Variational Autoencoder

The CVAE extends the VAE to learn $p_\theta(\mathbf{x} \mid \mathbf{c})$ instead of $p_\theta(\mathbf{x})$, where $\mathbf{c}$ is the condition vector. Both the encoder $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c})$ and the decoder $p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c})$ now depend on $\mathbf{c}$, while the prior is kept condition-independent, $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The same derivation, now carrying the condition $\mathbf{c}$ through (Appendix A.6), yields the conditional ELBO:

$$\log p_\theta(\mathbf{x} \mid \mathbf{c}) \geq \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c})} [\log p_\theta(\mathbf{x} \mid \mathbf{z}, \mathbf{c})] - \text{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}) \parallel p(\mathbf{z})) \quad (2.16)$$

Conditioning by concatenation. The condition $\mathbf{c}$ is fed into both networks by concatenation: the encoder takes $[\mathbf{x}, \mathbf{c}]$ as input, the decoder $[\mathbf{z}, \mathbf{c}]$. No separate conditioning architecture is used. This is the standard CVAE pattern and matches the implementations of Barmada et al. [3] as well as Fang and Ji [12]. Figure 2.9 contrasts the two architectures at training time: the only structural difference is the condition $\mathbf{c}$, which is concatenated to both encoder and decoder inputs in the CVAE.

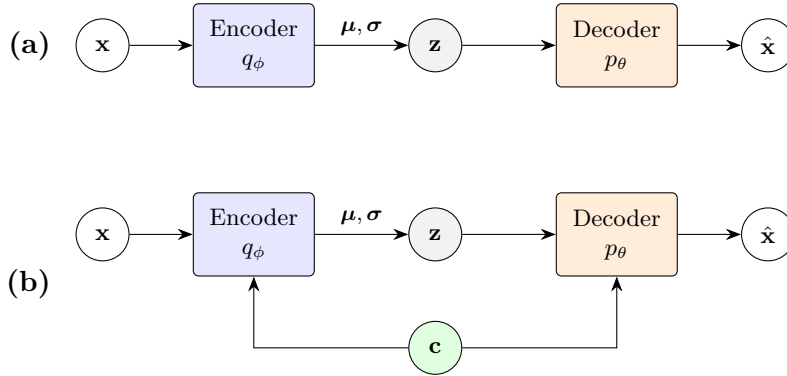


FIGURE 2.9: Training-time architectures of (a) the VAE and (b) the CVAE. In both cases the encoder maps the input to a Gaussian posterior with parameters (μ, σ) , the latent $\mathbf{z}$ is obtained by the reparameterisation trick (Figure 2.8), and the decoder reconstructs $\hat{\mathbf{x}}$.

Latent space as residual. With $\mathbf{c}$ already supplied to both networks, the encoder no longer needs to encode the variation in $\mathbf{x}$ that is determined by the condition. The latent $\mathbf{z}$ instead captures the residual variation in $\mathbf{x}$ *within* the set $\mathcal{G}(\mathbf{c})$ of geometries compatible with the requirements. At inference time this is the desired behaviour: $\mathbf{c}$ is fixed by the client requirement while $\mathbf{z}$ is sampled repeatedly from the prior, producing a diverse set of compliant candidates from one forward pass per sample.

Figure 2.10 illustrates this inference setting: the encoder of Figure 2.9(b) is dropped, and the decoder is driven directly by samples from the prior together with the fixed condition.

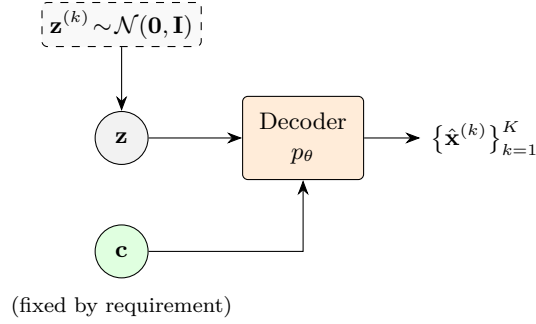


FIGURE 2.10: CVAE at inference time. The encoder of Figure 2.9(b) is dropped: $\mathbf{c}$ is fixed at the design requirement and K prior samples $\mathbf{z}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ drive the decoder, yielding K diverse candidate geometries in one forward pass each.

Output activation. The decoder ends in a sigmoid, constraining each component of $\hat{\mathbf{x}}$ to $[0, 1]$. The geometry parameters of Table 1.1 are min-max normalised to the same interval (Chapter 5), so the sigmoid output is in correspondence with the physical parameter ranges by construction.

Differentiability of the Decoder. Both the encoder and decoder are MLPs of the form introduced in §2.2. Because the decoder output is a differentiable function of $\mathbf{z}$ and of the network parameters θ , any scalar loss on $\hat{\mathbf{x}}$ back-propagates through it without modification. This makes it possible to insert an external, performance-evaluating network directly into the training objective: gradients flow from the performance penalty through the decoder into its weights, shaping the generated geometries towards compliance. The concrete realisation (using the frozen surrogate of Chapter 4 as a differentiable penalty term in the CVAE loss) is developed in Chapter 5.

Alignment with the framework. With the model fully described, three properties make the CVAE particularly well suited to the inverse-design task pursued in this work. First, the ELBO of §2.3.2 gives an explicit, likelihood-based training signal, which is easier to tune than the adversarial objective of a GAN. Second, the decoder is a plain, differentiable MLP: as the previous paragraph set out, this is what allows a performance penalty to be carried *inside* the training loss, so that compliance can be pursued while the geometries are being learned. Third, a candidate geometry is produced in a single forward pass of the decoder, so many compliant designs can be drawn for the same requirement vector almost instantly. Together, these three properties make the CVAE a better fit for this thesis than the otherwise defensible GAN and diffusion alternatives.

2.4 Network Design and Training Techniques

The surrogate MLP (§2.2, Chapter 4) and the CVAE (§2.3, Chapter 5) share many of the same choices in how they are built and trained, though a few differ between the two. This section gathers these choices; each technique is explained in more detail in Appendix A.5, and the specific values used are reported in Chapters 4 and 5.

Both networks use the SiLU activation [17], which gave more consistent convergence than ReLU in preliminary experiments. The surrogate uses the Huber loss [21] on its compressed targets to dampen the influence of poorly conditioned components; the CVAE uses the mean squared error on standardised geometry parameters as the reconstruction term inside its training objective. Both are trained with Adam [22] as the optimiser. Generalisation is controlled by early stopping on the validation loss; the learning rate follows a plateau-based reduction schedule for the surrogate and a cosine-annealing schedule [30] for the CVAE. Gradient clipping [38] and layer normalisation [2] are applied to the CVAE, whose composite training objective occasionally produces large transient gradients and benefits from the more stable per-sample normalisation. Residual connections [16] are used in the surrogate, allowing several blocks to be stacked while keeping training tractable; the CVAE encoder and decoder use plain MLP layers without skips.

2.5 AI-Based Electric Machine Design

Deep learning has become a prominent tool for accelerating the finite-element evaluation that drives optimisation in electric-machine design. The bibliometric study of Do Nascimento et al. [11], surveying 246 high-impact articles, confirms that surrogate-assisted multi-objective optimisation is now a dominant methodology for permanent-magnet, induction, and reluctance machines. Parekh, Flore and Schöps [37] provide a prominent example: they train a VAE–DNN combination that jointly predicts cost, maximum power, and maximum torque KPIs and generates new designs across a unified latent space spanning both PMSM and asynchronous machines; a single FE evaluation of a PMSM costs hours, whereas the trained meta-model returns a prediction in negligible time. Di Barba, Mognaschi and Wiak [9] generalise this to neural metamodeling of EM fields, with convolutional networks acting as equation-less forward and inverse solvers and a multi-fidelity training scheme that cuts the FE budget by roughly 30 %.

A parallel body of work turns to generative models that produce motor designs directly. Akbari and Lowther [1] take a GAN-based approach: their CDC-GAN generates six geometric parameters of axial-flux permanent-magnet machines, conditioned on a single scalar KPI such as average torque used as a class label, with composite distribution-matching, correlation, and diversity losses. Architecturally the closest precedent in the EM literature is Barmada et al. [3]: a convolutional CVAE reconstructs full magnetic field maps from partial subdomain observations

and is paired with a downstream CNN for source identification, with latent-space optimisation by automatic differentiation refining the prediction at inference time. The closest structural precedents lie outside electric-machine design. Yang, Lee and Yee [47] couple a VAE that generates target pressure distributions with an MLP that predicts shape parameters and aerodynamic quantities of interest, applied to the inverse design of wind-turbine airfoils. Fang and Ji [12] demonstrate a similar CVAE-for-inverse-design recipe in de novo molecular design. Fu et al. [13] explore the conditional-diffusion approach: their iterative denoising procedure achieves high-fidelity synthesis but at an inference cost that grows with the number of denoising steps.

The recent review of Tanveer et al. [45] situates this family of approaches within the broader generative-design landscape. A separate question concerns how feasibility constraints are enforced. Nguyen and Donti [36] integrate a feasibility-seeking projection step that minimises constraint violations and provides formal guarantees; Di Lorenzo et al. [10] embed physical relations directly in a graph-neural-network architecture. Hard-constraint methods of this kind guarantee feasibility but introduce inference-time overhead and complicate end-to-end training.

Positioning of this work

Within this landscape, the framework developed in the remainder of this thesis combines three choices that, taken together, set it apart from the surveyed precedents. First, where the closest generative precedents in electric-machine design condition on a single scalar KPI [1] or operate on field maps [3], the CVAE developed here generates 17 geometric parameters of the double-V IPMSM (Table 1.1) directly, conditioned on the multi-component requirement vector $\mathbf{C}$ of §3.2, which encodes two operating points and the battery and inverter limits. Second, Chapter 5 deliberately departs from the standard generate-then-verify pipeline by coupling the decoder to the frozen performance surrogate of Chapter 4 as a differentiable regulariser *inside* the training loss. This requires the surrogate to be both differentiable and accurate enough to act as a training signal rather than only as a downstream check, and it lets performance pressure shape the geometries while they are being generated rather than afterwards. Third, Chapter 6 extends this differentiable chain to gradient-based latent-space optimisation under joint performance and structural-feasibility penalties, a combination the author has not encountered in the surveyed electric-machine-design literature.

Chapter 3

Dataset Construction and Performance Abstraction

This chapter describes how the dataset that drives the surrogate of Chapter 4 and the generative model of Chapter 5 is constructed. Because no required performance profile is known a priori per sample, the conventional ZONDA workflow of §2.1.3 is inverted: the winding ratio and electrical limits are sampled jointly with the geometry, and the full performance profile is then reduced to the two discrete operating points that finalise the condition vector $\mathbf{C}$.

3.1 Data Generation Pipeline

The ZONDA Toolchain executes the geometry-to-performance pipeline of Figure 2.5 in the forward direction: a candidate geometry, a winding configuration, and a set of electrical limits are evaluated to yield a performance profile, and the toolchain’s optimisation loop iterates this evaluation against a client-specified target PP. The data-generation problem of this thesis is different. Here the goal is to construct paired examples $(\mathbf{G}, \mathbf{C})$ such that the geometry $\mathbf{G}$ realises the condition vector $\mathbf{C}$, and there is no client-specified target PP attached to any single sample. The required PP for a given sample is therefore not an input but a *computed* consequence of the sampled inputs.

This has the following consequence for the workflow.

The optimal winding ratio is no longer well-defined. In the conventional setting, N is selected per sample to align the field-weakening transition with the requested base speed (§2.1.3). Without a target PP, no such alignment criterion exists. Instead, N must be sampled jointly with $\mathbf{G}$, I_{max} , and U_{max} , and the resulting PP simply reflects whichever value was drawn. This randomisation of N does not compromise coverage at inference time: because N takes only the four discrete values $\{1, 2, 3, 4\}$, any client-specified set of operating-point and electrical requirements can simply be queried four times (once per value of N) and the union of the four resulting candidate sets exhausts the discrete winding-configuration space.

Latin Hypercube Sampling

The 20-dimensional input space (17 free geometric parameters, I_{max} , U_{max} , and N) is sampled using Latin Hypercube Sampling (LHS), introduced by McKay, Beckman and Conover [33]. For a budget of M samples and d input variables with prescribed ranges, the procedure partitions the range of each variable into M equally probable bins, draws exactly one value from each bin per variable, and then pairs the per-variable draws through an independent random permutation to form M joint samples. By construction, every one-dimensional marginal is perfectly stratified: each bin contains exactly one sample. This is in contrast to plain Monte Carlo sampling, where individual marginals can exhibit visible clustering and gaps at moderate M . LHS does not enforce joint d -dimensional space-filling (an unlucky permutation can still leave regions of the joint space sparsely covered) but with $M \geq 10^4$ and $d = 20$ the residual joint clustering is small. The discrete winding ratio is sampled by partitioning the unit interval into four equal bins and mapping each draw to the corresponding value of N .

Physical feasibility filter

Because the geometry, the electrical limits, and the winding ratio are sampled independently, a non-negligible fraction of the LHS iterations leads to combinations that are not physically realisable: the cross-section of copper available in the stator slots is simply too small to carry the sampled RMS phase current at the sampled winding ratio. The relevant quantity is the slot current density,

$$J_{slot} = \frac{I_{max} \cdot N}{S_{slot}} \leq 30 \frac{\text{A}}{\text{mm}^2}, \quad (3.1)$$

where S_{slot} is the copper cross-section per slot fixed by the sampled stator geometry, I_{max} is the sampled RMS phase current, and N is the winding ratio. Equation (3.1) is simply equation (2.11) rearranged for J_{slot} , with the inequality enforcing the thermal limit on the resulting density. Ohmic dissipation in the copper scales with J_{slot}^2 , so above a certain current density the heat generated under sustained current can no longer be properly evacuated, and the conductor temperature exceeds the thermal rating of its insulation. The 30 A/mm² threshold is a limit enforced by Punch Powertrain for this type of machines. Removing infeasible samples is an important step: retaining physically impossible configurations would contaminate the training set with inputs that no real motor can realise, degrading the quality of what the models learn. In this sense, the value of the dataset depends as much on the quality of its samples as on the number of samples.

Resulting data generation workflow

The complete pipeline is summarised in Figure 3.1.

The FEM flux-map step in the conventional workflow (see Figure 2.5) is replaced by the neural-network surrogate of [8], which yields each (ϕ_d, ϕ_q) pair in milliseconds; the flux-linkage scaling by N and the PP construction itself are retained unchanged.

Each LHS iteration that survives the feasibility filter contributes a single dataset entry consisting of the 17 free geometric parameters, the electrical limits (I_{max}, U_{max}) , the winding ratio N , and the full computed performance profile, sampled at 250 rpm intervals from 250 rpm to 20 000 rpm.

The performance profile attached to each sample reflects the purely electromagnetic behaviour of the design under the sampled electrical limits and winding ratio, evaluated over the full speed range. Mechanical constraints (in particular the centrifugal-stress limit on the rotor laminations and magnet bridges) are not enforced during data generation.

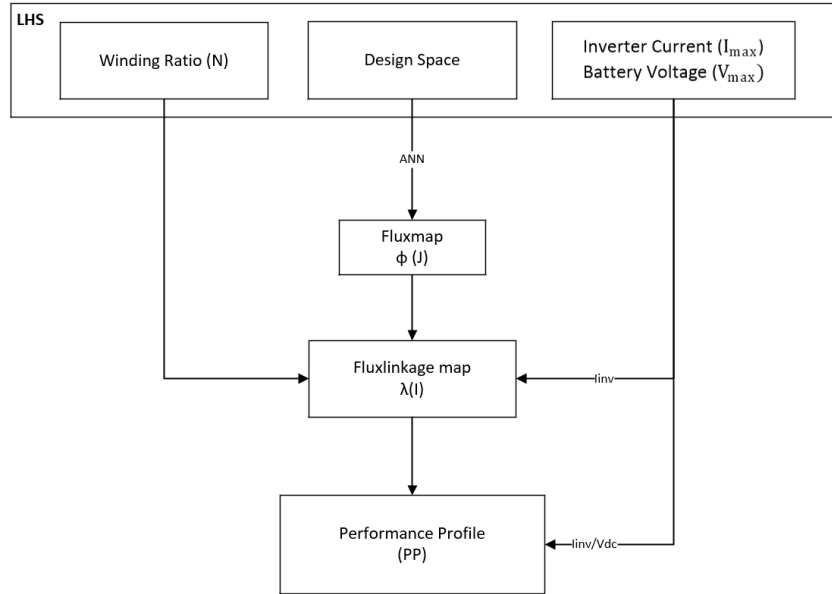


FIGURE 3.1: Data-generation workflow used in this work. Each iteration produces one $(\mathbf{G}, I_{max}, U_{max}, N, PP)$ data sample.

3.2 Operating Points and Condition Vector

The dataset contains, for each sample, a full performance profile sampled at 250 rpm resolution. Using this profile directly as a conditioning signal is impractical: it is high-dimensional and not the form in which client requirements are issued. In Punch Powertrain’s industrial workflow, a client specifies a target performance through a small set of operational points: peak torque at base speed, peak torque at top speed, and additionally some continuous operating conditions (not considered in this work, see §1.1.1). This section reduces the full performance profile to two scalars and, in doing so, finalises the condition vector $\mathbf{C}$.

Choice of operating points

The profile is summarised by two operating points (Figure 3.2):

- $OP1 = (\omega_{OP1}, T_{OP1})$, the peak-torque condition at base speed, characterising the constant-torque region;
- $OP3 = (\omega_{OP3}, T_{OP3})$, the peak-torque condition at top speed. Concretely, ω_{OP3} is taken as the lowest speed at which the computed peak-torque envelope reaches 0 Nm, or capped at 20 000 rpm if the envelope has not yet decayed to zero by that speed.

These endpoints are sufficient to assess the compliance of a candidate geometry against a client’s specification.

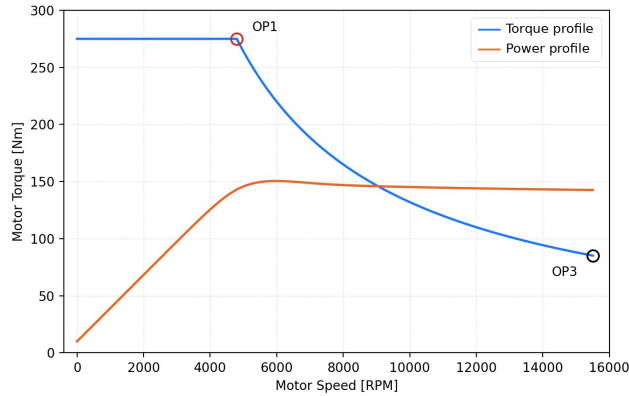


FIGURE 3.2: Reduction of the full performance profile to the two operating points OP1 and OP3.

Final condition vector

Combining the two operating points with the electrical limits I_{max} and U_{max} and the winding ratio N (§2.1.3), the condition vector becomes

$$\mathbf{C} = [\omega_{OP1}, T_{OP1}, \omega_{OP3}, T_{OP3}, I_{max}, U_{max}, N] \in \mathbb{R}^7, \quad (3.2)$$

or, in compact form, $\mathbf{C} = [\text{OP1}, \text{OP3}, I_{max}, U_{max}, N]$. This is the form adopted throughout Chapters 4–6.

Electrical limits convention

Client requirements enter as a DC-bus voltage and a maximum RMS phase current, $(U_{DC,nom}, I_{inv,max,RMS})$. The current limit is stored directly, $I_{max} = I_{inv,max,RMS}$. The voltage limit is converted to the maximum RMS phase voltage the inverter can deliver under SVPWM [19],

$$U_{max} = U_{ph,RMS,max} = \frac{M_{max}}{1.15\sqrt{6}} U_{DC,nom}, \quad (3.3)$$

with M_{max} the maximum modulation index. Both I_{max} and U_{max} in $\mathbf{C}$ are therefore RMS phase quantities (units A_{rms}, V_{rms}).

Compliance criterion

A sampled geometry is declared compliant with a target $\mathbf{C}$ when its computed PP satisfies the torque demand at both operating points,

$$T_{sample}(\omega_{\text{OP1}}) \geq T_{\text{OP1}} \quad \text{and} \quad T_{sample}(\omega_{\text{OP3}}) \geq T_{\text{OP3}}. \quad (3.4)$$

Figure 3.3 illustrates this two-point compliance check on representative examples.

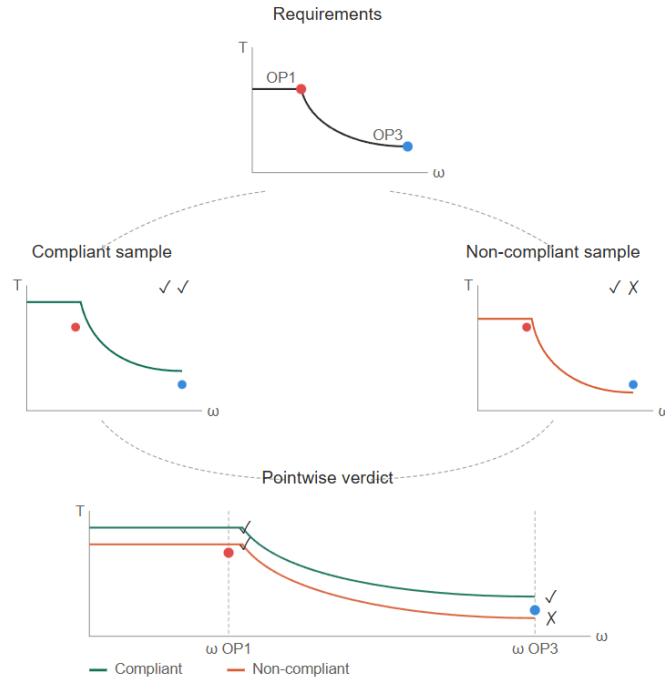


FIGURE 3.3: Compliance through two-point evaluation. A sample is accepted iff its computed PP lies above the requirement at both ω_{OP1} and ω_{OP3} .

3.3 Dataset and Preprocessing

3.3.1 Exploratory Analysis of the Dataset

The data generation pipeline was applied to 10^5 LHS samples. In deep learning, larger datasets generally improve performance by exposing the model to a wider range of inputs. No strict minimum exists, but given the 24-dimensional input space and the complexity of the generative task, a dataset on the order of 10^5 samples is a sensible target. After the physical feasibility filter, 85 384 samples survive (an acceptance rate of $\approx 85\%$). This post-filter set is the dataset used for all results reported in the remainder of this work. A statistical summary per variable is reported in Table 3.1 (divided into geometric parameters, electrical limits & winding ratio, and performance requirements); a printout of representative **G** and **C** rows can be found in Appendix B.

TABLE 3.1: Statistical summary of the dataset after physical feasibility filtering ($M = 85\,384$ samples). Geometry-parameter symbols are those of Table 1.1.

Variable	Unit	Avg.	Std Dev	Min	25%	50%	75%	Max
DSO	m	0.2169	0.0200	0.1800	0.2001	0.2178	0.2343	0.2500
GAP	mm	0.849	0.202	0.500	0.674	0.849	1.024	1.200
KAR	–	0.750	0.202	0.400	0.575	0.749	0.924	1.100
KBETA	–	0.275	0.043	0.200	0.238	0.275	0.313	0.350
KFD	–	1.900	0.404	1.200	1.550	1.900	2.250	2.600
KGAMMA	–	0.175	0.072	0.050	0.113	0.175	0.237	0.300
KMQ1	–	0.650	0.144	0.400	0.525	0.650	0.775	0.900
KMQ2	–	0.650	0.144	0.400	0.525	0.650	0.775	0.900
KYH	–	0.496	0.057	0.400	0.447	0.495	0.545	0.600
KPQ1	–	0.500	0.231	0.100	0.300	0.500	0.700	0.900
KPQ2	–	0.500	0.231	0.100	0.300	0.500	0.701	0.900
KQI	–	0.650	0.087	0.500	0.575	0.650	0.725	0.800
KQIVAR	–	0.425	0.072	0.300	0.363	0.425	0.488	0.550
KR0	–	0.625	0.043	0.550	0.587	0.625	0.662	0.700
KR1	–	0.500	0.115	0.300	0.400	0.500	0.600	0.700
KSR	–	0.722	0.043	0.650	0.685	0.721	0.759	0.800
KTW	–	0.497	0.058	0.400	0.447	0.495	0.546	0.600
I_{max}	A_{rms}	440.4	160.4	180.0	302.1	429.0	572.5	750.0
U_{max}	V_{rms}	165.1	58.4	63.9	114.6	165.0	215.6	266.2
N	–	2.31	1.08	1	1	2	3	4
ω_{OP1}	rpm	6 824	5 483	250	2 750	5 000	9 500	20 000
T_{OP1}	Nm	337.4	231.9	14.1	164.8	279.8	452.6	1 746.1
ω_{OP3}	rpm	19 541	2 062	1 250	20 000	20 000	20 000	20 000
T_{OP3}	Nm	67.6	45.7	1.0	34.2	58.2	91.6	275.5

Table 3.1 reports statistics only for the two abstracted operating points OP1 and OP3; each sample in fact contains the full performance profile at 250 rpm resolution from 250 rpm to 20 000 rpm. These full profiles, not the two-point abstractions, are the targets used to train the MLP surrogate of Chapter 4.

Figure 3.4 shows how OP1 and OP3 are distributed across the torque–speed plane in the working dataset. Each pair of points (red for OP1 and blue for OP3) represents one sample. For reference, the left panel shows an earlier and independent exploratory run of 10^3 LHS samples (845 post-filter), included to make the density gain of the full 10^5 -sample run visible.

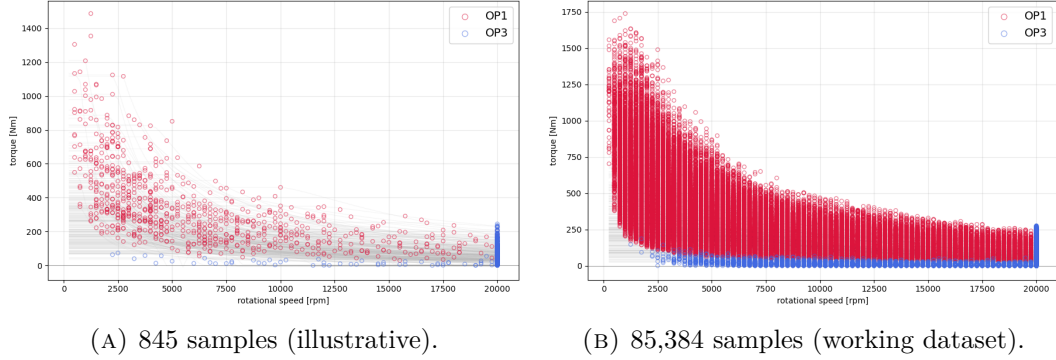


FIGURE 3.4: Distribution of OP1 and OP3 in the torque–speed plane. Left: exploratory 10^3 -sample run (845 post-filter). Right: working dataset (85 384 post-filter samples).

Beyond the per-variable marginals reported in Table 3.1, it is also useful to examine how the variables of the dataset relate to one another. Quantifying these relationships requires the choice of a correlation measure, which is made first; the joint structure of the dataset is then visualised in a pair-matrix.

Correlation Measure

The most common choice, the Pearson coefficient r , quantifies only the strength of a linear association between two variables. It is most informative when the data is approximately elliptically distributed (e.g. bivariate normal); outside that regime, two strongly dependent variables can still yield $r \approx 0$, and a perfectly monotone but non-linear relation can yield $|r| < 1$ [15]. Pearson r is also sensitive to outliers. In the present context the dependencies between geometric parameters and electromagnetic performance quantities (torque-speed envelope) are governed by a strongly non-linear physics involving magnetic saturation, reluctance effects and field-weakening, so a linear measure is unlikely to capture them faithfully.

The Spearman coefficient ρ_S is the Pearson coefficient computed on the ranks of the observations and therefore measures the degree of any monotone relationship; it is invariant under monotone transformations of either variable and more robust to outliers, which makes it the more appropriate choice here:

$$\rho_S = \text{cor}(\text{rank}(x), \text{rank}(y)) = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}, \quad d_i = \text{rank}(x_i) - \text{rank}(y_i), \quad (3.5)$$

where the closed-form expression on the right holds when all ranks are distinct.

Graphical Summary

With the correlation measure fixed, the joint structure of the dataset can be inspected. Since a full 24×24 matrix over all geometry parameters and conditions would be unreadable, a 10×10 pair-matrix is presented in Figure 3.5, over the seven components of the condition vector $\mathbf{C}$ together with the three geometry parameters most strongly correlated with $\mathbf{C}$: KAR, DSO and KMQ1. This geometry selection is supported by a 7×17 Spearman heatmap reported in Figure B.1 in Appendix B.

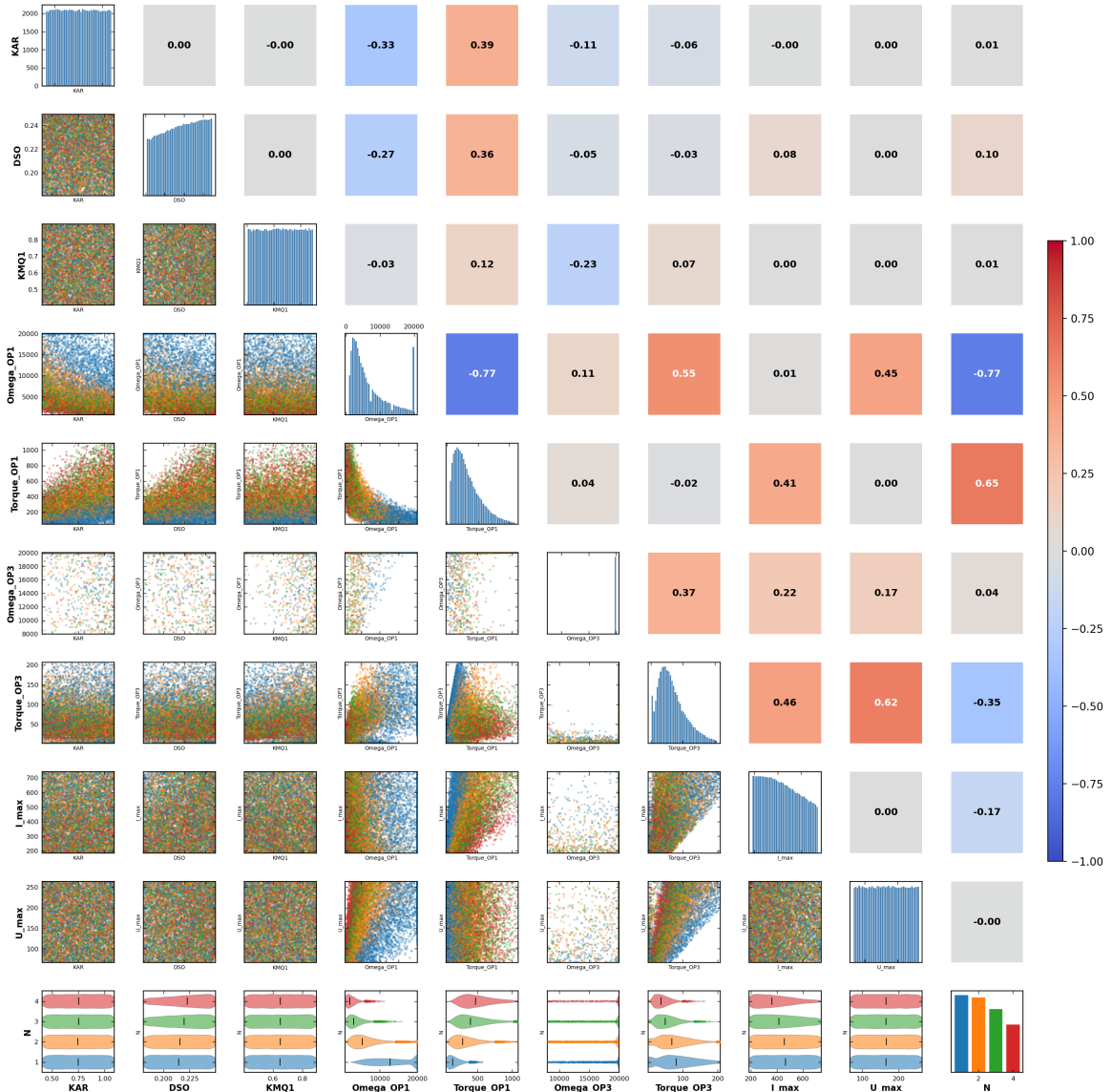


FIGURE 3.5: Pair-matrix of the condition vector $\mathbf{C}$ and the three geometry parameters most strongly correlated with $\mathbf{C}$ (KAR, DSO, KMQ1).

The diagonal of the matrix carries the marginal histogram of each variable, with coloured bar counts for the discrete winding ratio N (for all values of N). The lower triangle shows scatter plots of a 10^4 -sample subset that reveal the joint distribution between each pair of variables, with points coloured by N ; cells with N on either axis show per-category violin plots instead. The upper triangle reports the Spearman correlations ρ_S on a coloured $[-1, 1]$ background.

The strongest correlations in the upper triangle are physically expected: $\rho_S(\omega_{\text{OP1}}, T_{\text{OP1}}) = -0.77$ reflects the fact that OP1 sits at the corner of the envelope, where the product $T_{\text{OP1}} \cdot \omega_{\text{OP1}}$ is bounded by the inverter’s apparent power $\sim U_{\text{max}} I_{\text{max}}$, so across samples the corner traces an approximately iso-power locus $T \propto 1/\omega$; $\rho_S(\omega_{\text{OP1}}, N) = -0.77$ follows from the $\omega \propto 1/N$ scaling of the field-weakening transition, and $\rho_S(T_{\text{OP1}}, I_{\text{max}}) = 0.65$ follows from $T \propto I$ at fixed geometry. The diagonal reveals a separate observation about the marginals: although the LHS input is uniformly stratified per variable, the post-filter marginals are not. The bins $N = 3$ and $N = 4$ are visibly trimmed by the feasibility filter, and so are high I_{max} and low DSO. This is a direct consequence of the slot current-density limit (3.1): the product $N \cdot I_{\text{max}}$ enters the numerator and is exactly what drives infeasible samples out, while a low DSO reduces S_{slot} and therefore tightens the bound. For completeness, the corresponding 17×17 pair-matrix between all geometry parameters is included as Figure B.2 in Appendix B; it shows $|\rho_S| < 0.05$ throughout, confirming that the LHS marginals remain effectively independent.

3.3.2 Dataset Partitioning

The dataset is split into 70 % training, 15 % validation, and 15 % test, with the role of each partition as defined in §2.2. The same fixed-seed random split is reused by the surrogate of Chapter 4 and the CVAE of Chapter 5, so both models are evaluated on identical samples.

Chapter 4

Surrogate MLP for Geometry-to-Performance Mapping

This chapter introduces the neural surrogate that replaces, at inference time, the ZONDA performance-profile (PP) computation. The motivation is straightforward: even after replacing the FEM flux-map step with the surrogate of [8] (§3.1), assembling a full PP for a candidate geometry still takes seconds per sample, which is prohibitive in the inner loop of the latent-space optimisation of Chapter 6. A small MLP trained on the dataset of Chapter 3 learns the geometry-to-PP map directly and evaluates a full curve in sub-millisecond time.

4.1 Network Inputs, Outputs and Architecture

Inputs

The surrogate takes as input the 17 variable geometric parameters $\mathbf{G}$ together with the inverter limits (I_{max}, U_{max}) and the winding ratio N , which are extracted from the condition vector $\mathbf{C}$. The input vector is therefore

$$\mathbf{x}_{\text{MLP}} = [\mathbf{G}, I_{max}, U_{max}, N] \in \mathbb{R}^{20}. \quad (4.1)$$

Knowledge of the geometric parameters alone is not sufficient to learn an adequate mapping from geometry to performance: the performance profile is shaped, in addition to $\mathbf{G}$, by the electrical limits and the winding ratio, as established in §2.1.2 and §2.1.3. Without these extra inputs, the surrogate would be asked to predict, from $\mathbf{G}$ alone, a target that is not a function of $\mathbf{G}$ alone.

Input normalisation. A per-feature z -score is applied to $\mathbf{x}_{\text{MLP}}$ before it enters the network:

$$\tilde{\mathbf{x}}_{\text{MLP}} = \frac{\mathbf{x}_{\text{MLP}} - \boldsymbol{\mu}_x}{\boldsymbol{\sigma}_x}, \quad (4.2)$$

with μ_x and σ_x the per-feature mean and standard deviation estimated on the training partition and stored as an artefact, so that inference reuses the same statistics. A z -score is used because the 20 features span very different physical magnitudes (geometric parameters, currents, voltages and the integer winding ratio) and a z -score puts them on a comparable numerical scale.

Outputs

The target is the full performance profile at the 80 RPM samples introduced in §3.1,

$$\mathbf{y} = [T(\omega_1), T(\omega_2), \dots, T(\omega_{80})] \in \mathbb{R}^{80}, \quad \omega_i = 250 \cdot i \text{ rpm.} \quad (4.3)$$

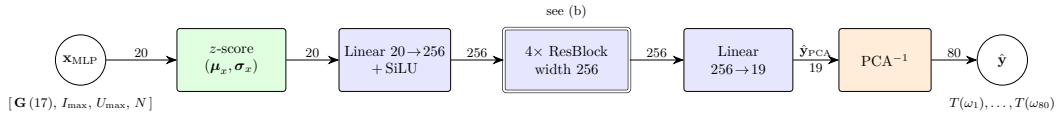
The 80 components are strongly correlated along the speed axis, and learning all 80 of them independently is wasteful; Section 4.2 describes how $\mathbf{y}$ is compressed to a much smaller target before training.

Architecture

The network is a residual MLP. The 20-D normalised input is lifted to a 256-D hidden representation, processed through four residual blocks of width 256, and projected to the compressed output dimension. Each residual block applies two Linear-SiLU layers and adds the block’s input to the result, in the spirit of the residual connections introduced by [16]:

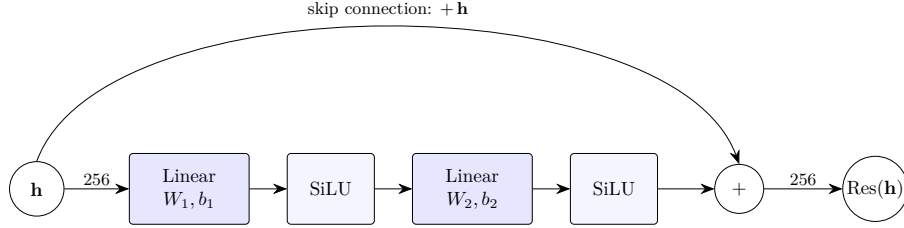
$$\text{Res}(\mathbf{h}) = \mathbf{h} + \text{SiLU}(W_2 \text{SiLU}(W_1 \mathbf{h} + b_1) + b_2). \quad (4.4)$$

The activation throughout is SiLU (the Sigmoid Linear Unit, $\text{SiLU}(x) = x \sigma(x)$); no dropout or batch normalisation is used. Figure 4.1 schematises the full forward pass together with the internal structure of one residual block. The full network has approximately 5.4×10^5 trainable parameters.



(A) Forward pass. The 20-D geometric and electrical input is z -score normalised, lifted to 256 channels, processed by four residual blocks (double border indicates repetition; internal structure shown in (b)), projected onto the 19-D PCA basis, and inverse-transformed to the 80-point torque curve.

FIGURE 4.1: Network architecture of the surrogate MLP. Blue blocks are trainable; the green (z -score) and orange (PCA^{-1}) blocks are deterministic, fitted-once transformations exported alongside the network weights.



(B) Internal structure of one residual block: two Linear–SiLU layers in series with an additive skip connection from the block input, cf. Eq. (4.4).

FIGURE 4.1: Network architecture of the surrogate MLP (continued).

4.2 Output-space Compression by Principal Components

Motivation

The 80 components of $\mathbf{y}$ in (4.3) sample a smooth, physically constrained torque–speed envelope: across the dataset, the variation along the speed axis is largely explained by a handful of envelope shapes (constant-torque plateau, field-weakening hyperbola, position of the corner point). Regressing the 80 RPM values independently forces the network to relearn this correlation structure for every sample. Two practical consequences follow: a larger output layer than warranted by the intrinsic dimensionality of the target manifold, and a loss that is dominated by redundant per-RPM errors rather than by genuinely independent error modes.

Procedure

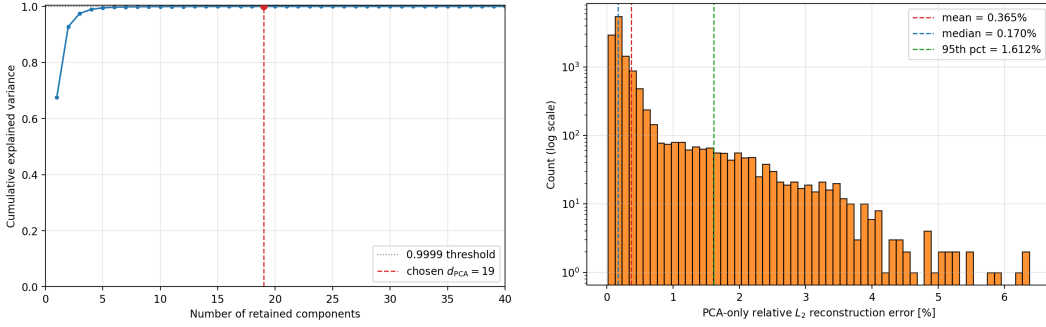
Principal-component analysis (PCA) is fitted on the training-set targets $\mathbf{Y}_{\text{train}} \in \mathbb{R}^{M_{\text{train}} \times 80}$ with a cumulative-variance criterion of 0.9999. This retains $d_{\text{PCA}} = 19$ components, an $80 \rightarrow 19$ reduction. Training targets, validation targets and test targets are all transformed by the same fitted PCA. The network’s output is therefore in the 19-D PCA basis; predictions are inverse-transformed to the 80-D torque–curve space before any physical metric is computed:

$$\mathbf{y} \xrightarrow{\text{PCA}} \mathbf{y}_{\text{PCA}} \in \mathbb{R}^{19} \xrightarrow{\text{network}} \hat{\mathbf{y}}_{\text{PCA}} \xrightarrow{\text{PCA}^{-1}} \hat{\mathbf{y}} \in \mathbb{R}^{80}. \quad (4.5)$$

The fitted PCA transformation is exported alongside the network weights and the z -score statistics.

Characterisation of the PCA component

Two diagnostics are reported. First, the cumulative explained-variance curve (Figure 4.2a) confirms the rapid concentration of variance in the leading components and motivates the choice of 19. Second, because the PCA introduces its own reconstruction error that is independent of the network, the *PCA-only* reconstruction error $\frac{\|\mathbf{y} - \text{PCA}^{-1}(\text{PCA}(\mathbf{y}))\|_2}{\|\mathbf{y}\|_2}$ is reported over the test set (Figure 4.2b). This isolates the ceiling on accuracy imposed by the compression itself, an irreducible error floor separate from the additional error introduced by the regressor.



(A) Cumulative explained variance against the number of retained components. The chosen $d_{\text{PCA}} = 19$ is marked. (B) Distribution of the per-sample PCA-only relative L_2 reconstruction error on the test set.

FIGURE 4.2: Characterisation of the output-space PCA.

4.3 Training Configuration and Convergence

Loss and optimiser

The training loss is the Huber loss [21] on the PCA-compressed targets,

$$\mathcal{L}_\delta(\hat{\mathbf{y}}_{\text{PCA}}, \mathbf{y}_{\text{PCA}}) = \frac{1}{d_{\text{PCA}}} \sum_{k=1}^{d_{\text{PCA}}} \begin{cases} \frac{1}{2}(\hat{y}_k - y_k)^2 & \text{if } |\hat{y}_k - y_k| \leq \delta, \\ \delta (|\hat{y}_k - y_k| - \frac{1}{2}\delta) & \text{otherwise,} \end{cases} \quad (4.6)$$

which behaves quadratically near zero and linearly in the tails. This choice is motivated by the distribution of PCA-component values: the leading components capture envelope-shape variation that is widely distributed across the dataset, while higher-order components carry a small number of large-magnitude residuals from atypical curves. A pure MSE on the PCA basis would let those outliers dominate; the Huber loss attenuates their leverage without giving up smooth gradients in the bulk of the distribution.

The optimiser is Adam with an initial learning rate of 10^{-3} , batch size 256, and a maximum budget of 1000 epochs. A plateau-based schedule halves the learning rate whenever the validation loss fails to improve for 10 consecutive epochs (factor 0.5, patience 10), and training is terminated early when the validation loss fails to improve for 30 consecutive epochs.

Training-curve diagnostics

Figure 4.3 shows the training and validation Huber loss over the course of training. Both curves descend together with a stable train–validation gap that does not widen during training; in the final plateau the validation loss settles at roughly $1.8\times$ the training loss, with no progressive overfitting signature. The best validation loss is reached at epoch 394 (marked in green on the figure), after which the patience counter triggers and the run terminates shortly thereafter. The plateau scheduler steps the learning rate down whenever the validation loss flattens; the effect is most visible around epochs 60, 120 and 200, where the validation curve is pulled distinctly downwards after each reduction. After the last effective reduction both curves flatten at their plateau values and the patience criterion terminates training before any further refinement is possible.

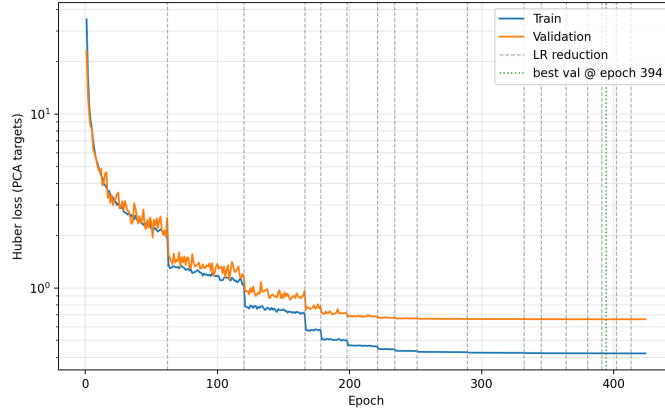


FIGURE 4.3: Training (blue) and validation (orange) Huber loss versus epoch. Vertical dashed lines mark learning-rate reductions triggered by the plateau scheduler; the run is terminated by the early-stopping criterion shortly after the best validation loss is reached.

4.4 Test-set Accuracy on the Denormalised Performance Profile

All metrics in this section are computed on the held-out test partition after inverse-PCA, i.e. directly on the 80-point torque curve in physical units. This evaluates the surrogate on what it is ultimately meant to predict, and puts the errors on the same scale as the PCA-only reconstruction baseline of §4.2. Two complementary views are reported: a single global table (Table 4.1) and three error-distribution figures (Figures 4.4–4.6).

Aggregate metrics

Let $\hat{\mathbf{y}}^{(n)}$ and $\mathbf{y}^{(n)}$ denote the predicted and reference torque curves for test sample n . The relative L_2 error per sample,

$$\varepsilon_n = \frac{\|\hat{\mathbf{y}}^{(n)} - \mathbf{y}^{(n)}\|_2}{\|\mathbf{y}^{(n)}\|_2 + \epsilon}, \quad (4.7)$$

together with its mean, median and a high-quantile summary, gives a curve-level quality measure that is insensitive to the absolute torque magnitude of the sample. Alongside, an unnormalised MAE (in Nm) and RMSE (in Nm) describe the absolute error, and a coefficient of determination R^2 averaged per sample describes the fraction of curve variance the surrogate explains. The MAPE is reported for completeness but, because samples contain low-torque RPM regions where small absolute errors translate into large percentage errors, it is not the headline metric.

TABLE 4.1: Test-set accuracy of the surrogate on the 80-point denormalised torque curve.

Metric	Value	Notes
Huber loss on PCA targets (test)	0.6512	training-domain reference
Global relative L_2 error	0.65 %	Frobenius over the full test set
Mean per-sample relative L_2	0.62 %	
Median per-sample relative L_2	0.41 %	
95 th -percentile per-sample L_2	2.01 %	
Worst-case per-sample relative L_2	9.55 %	
MAE (Nm)	0.708	
RMSE (Nm)	1.3735	inflated by low-torque RPM regions
Global MAPE	12.64 %	

Distribution of the error

Figure 4.4 is a parity plot of predicted versus reference torque, aggregated over every (sample, RPM) pair in the test set. A dense band along the identity line extends over the full torque range, from approximately 0 to 1700 Nm; the only visible departure is a broader scatter in the low-torque region ($\lesssim 200$ Nm), where small absolute residuals translate into relatively larger fractional errors and inflate the global MAPE entry of Table 4.1.

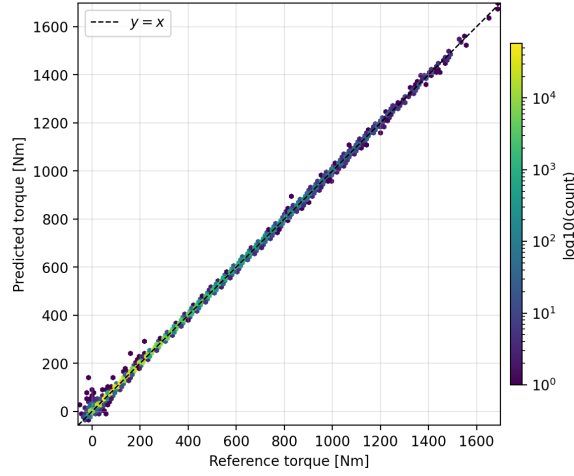


FIGURE 4.4: Parity plot of predicted versus reference torque over the entire test set, aggregated across all 80 RPM samples per curve.

Figure 4.5 reports the distribution of ε_n across test samples on a logarithmic vertical axis: the bulk of the test set falls below the 1 % fidelity threshold, with a mean of 0.62 %, a median of 0.41 % and a 95th percentile of 2.01 %, and a thin, sharply decaying tail extends to the worst-case 9.55 % sample.

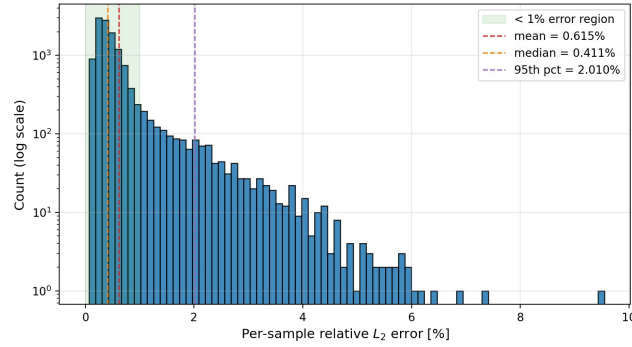


FIGURE 4.5: Distribution of the per-sample relative L_2 error (4.7) on the test set. Vertical lines mark the mean, the median and the 95th percentile.

Figure 4.6 shows four randomly chosen test samples with three curves each: the raw reference, the *PCA reference* (i.e. the raw reference compressed onto the 19-D PCA basis and inverse-transformed back, representing the best output the surrogate could in principle produce) and the surrogate prediction. In the first three panels the three curves visually overlap to within plotting resolution; the bottom-right panel instead shows a low-torque sample whose curve is nearly flat at around 80 Nm. There the PCA reference itself wobbles within roughly ± 0.3 Nm of the raw truth, and the network adds a comparable wobble: the absolute residual is tiny but its relative size dominates the high-quantile end of the error distribution. The gap between the raw reference and the PCA reference is the PCA’s own irreducible reconstruction error

characterised in §4.2; the gap between the PCA reference and the prediction is the residual attributable to the network itself.

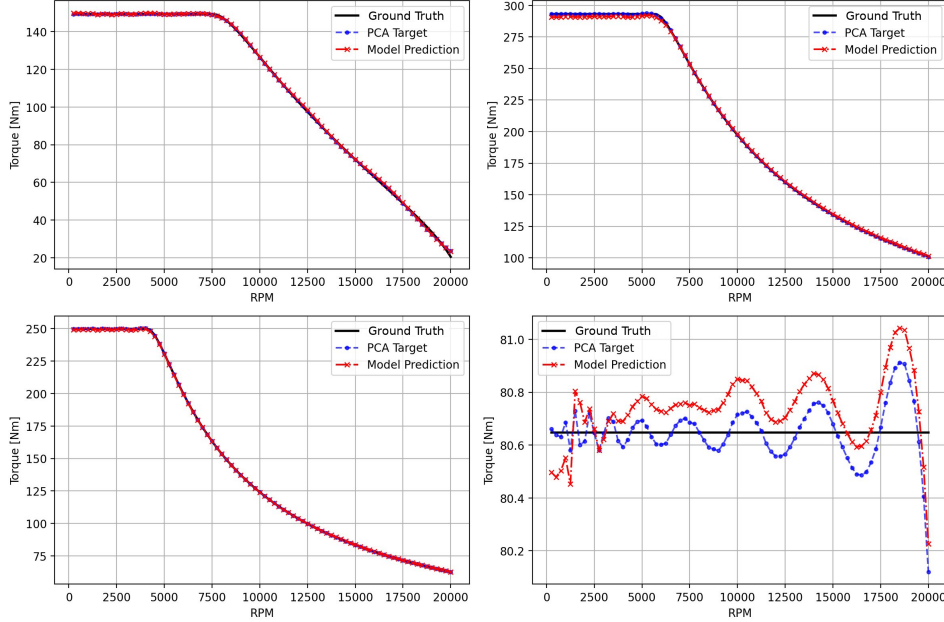


FIGURE 4.6: Four randomly chosen test samples. For each sample, the raw ZONDA reference (solid), the PCA reference (dashed) and the surrogate prediction (dash-dot) are overlaid. The gap between solid and dashed is the irreducible PCA reconstruction error of §4.2; the gap between dashed and dash-dot is the residual attributable to the network itself.

Surrogate-fidelity criterion

A surrogate-fidelity threshold of 1 % on the per-sample relative L_2 error is adopted for the remainder of this work: predictions below the threshold are considered accurate enough to participate, in place of a full FEM-based evaluation, in the compliance check of (3.4). The fraction of the test set that meets this threshold, reported in Figure 4.5, represents the rate at which the surrogate is expected to act as a faithful substitute.

4.5 Deployment within the Framework

The trained surrogate is exported as three coupled artefacts: the network weights, the z -score input statistics of (4.2), and the fitted PCA transform of §4.2. Together they define a fixed, deterministic mapping from a 20-D geometry-plus-electrical input to a denormalised 80-point torque curve. This frozen mapping is reused twice downstream: as a differentiable performance regulariser during CVAE training (Chapter 5) and as the in-loop performance evaluator inside the latent-space optimisation (Chapter 6), with the specifics developed in each chapter.

Chapter 5

Conditional Variational Autoencoder

5.1 Model Architecture

5.1.1 Encoder and Decoder Design

The encoder and decoder are symmetric three-layer MLPs with hidden widths (512, 256, 128) and (128, 256, 512) respectively, using SiLU activations and Layer-Norm after each linear layer. No dropout is used. The latent space is 15-dimensional. The encoder produces a mean $\boldsymbol{\mu}$ and a log-variance through two parallel linear heads; the log-variance head’s bias is initialised at -2.0 for early-training stability. Reparameterisation follows the standard form

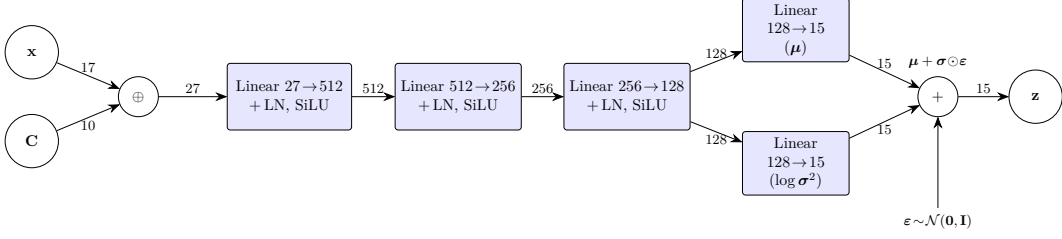
$$\mathbf{z} = \boldsymbol{\mu} + \exp\left(\frac{1}{2} \log \boldsymbol{\sigma}^2\right) \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (5.1)$$

The decoder’s output layer is a sigmoid, so generated geometries lie in $[0, 1]$ by construction. Figure 5.1 schematises both networks together with the reparametrisation path.

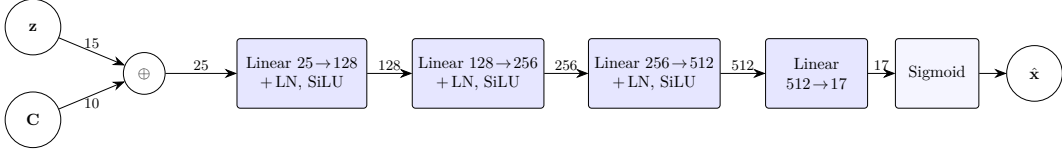
5.1.2 Conditioning and Output Constraints

The condition vector $\mathbf{C}$ is injected by plain concatenation to both the encoder input (with the normalised geometry $\mathbf{x}$) and the decoder input (with the latent $\mathbf{z}$). The discrete winding ratio $N \in \{1, 2, 3, 4\}$ is one-hot encoded before concatenation, so that the effective network input combines the normalised geometry $\mathbf{x} \in [0, 1]^{17}$ with the six continuous components of $\mathbf{C}$ (the (ω, T) pairs of operating points OP1 and OP3 together with $I_{\max}$ and $U_{\max}$, all z -scored) and the 4-dimensional one-hot encoding of N . Because the decoder output is sigmoid-activated, it is already bounded to $[0, 1]$, matching the min–max normalised geometry targets of §5.1.3.

5. CONDITIONAL VARIATIONAL AUTOENCODER



(A) Encoder. The geometry $\mathbf{x} \in [0, 1]^{17}$ and the condition $\mathbf{C} \in \mathbb{R}^{10}$ are concatenated and mapped through three Linear–LayerNorm–SiLU blocks of decreasing width. Two parallel heads produce $\boldsymbol{\mu}$ and $\log \sigma^2$; the reparametrisation $\boldsymbol{\mu} + \sigma \odot \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ yields the latent $\mathbf{z} \in \mathbb{R}^{15}$.



(B) Decoder. The latent $\mathbf{z} \in \mathbb{R}^{15}$ and the condition $\mathbf{C}$ are concatenated and processed by three Linear–LayerNorm–SiLU blocks of increasing width, mirroring the encoder. A linear output layer followed by a sigmoid produces the reconstructed geometry $\hat{\mathbf{x}} \in [0, 1]^{17}$.

FIGURE 5.1: Network architecture of the CVAE. Blue blocks are trainable; the lighter shade marks the deterministic sigmoid output. Dimension labels above each arrow indicate the tensor width at that stage, and concatenation of two tensors is denoted by $\oplus$.

5.1.3 Input Normalisation

The CVAE uses two per-feature normalisation schemes. The geometry vector $\mathbf{G}$ in physical units (Chapter 1) is min–max scaled per feature to its normalised counterpart $\mathbf{x}$,

$$\mathbf{x} = \frac{\mathbf{G} - \mathbf{G}_{\min}}{\mathbf{G}_{\max} - \mathbf{G}_{\min}} \in [0, 1]^{17}, \quad (5.2)$$

with bounds $(G_{\min,j}, G_{\max,j})$ obtained from the dataset min/max per feature (rounded outward to a resolution of 0.01) such that values are clipped to $[0, 1]$ after scaling. From this point onward $\mathbf{x}$ denotes the normalised CVAE target and $\hat{\mathbf{x}} = f_{\theta}(\mathbf{z}, \mathbf{C})$ the decoder output; inverting (5.2) maps $\hat{\mathbf{x}}$ back to a physical geometry $\mathbf{G}$.

The condition vector $\mathbf{C}$ is z -score standardised per feature with statistics fitted on the training partition. The same statistics are reused at inference time.

5.2 Standard CVAE

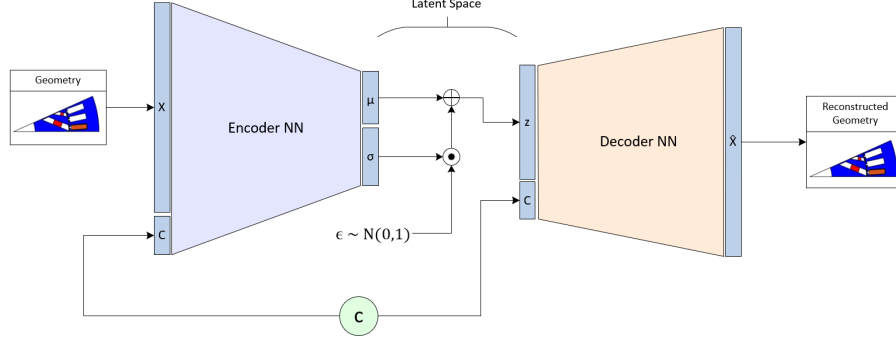


FIGURE 5.2: Training scheme of the standard CVAE.

5.2.1 Training Objective

The standard CVAE is trained on the β -weighted ELBO,

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{C})} [\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{C})] - \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{C}) \| p(\mathbf{z})), \quad (5.3)$$

with $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The reconstruction term is a mean-squared error between the normalised target $\mathbf{x} \in [0, 1]^{d_x}$ and the decoder output $\hat{\mathbf{x}}$. To prevent posterior collapse, a *free-bits* modification is applied to the KL term: the contribution of each latent dimension k is clamped to a minimum of $\tau = 0.1$ nats before averaging,

$$D_{\text{KL}}^{\text{FB}} = \frac{1}{d_z} \sum_{k=1}^{d_z} \max(D_{\text{KL},k}, \tau). \quad (5.4)$$

5.2.2 Training Setup

Optimisation uses Adam with a learning rate of 3×10^{-4} and a batch size of 128. The learning rate follows a cosine-annealing schedule ($T_{\text{max}} = 1000$, $\eta_{\text{min}} = 10^{-5}$), and gradients are clipped to a maximum norm of 1.0. The KL weight β is linearly annealed from 0 to $\beta_{\text{max}} = 0.05$ over the first 150 epochs and held constant thereafter. Training proceeds for up to 1000 epochs with early stopping on the validation loss (patience 75); the early-stopping counter is only activated after the β -warmup has finished. The dataset of Chapter 3 is split 70/15/15 into train/validation/test sets, using the input and target normalisation of §5.1.3.

5.3 Performance-Regularised CVAE

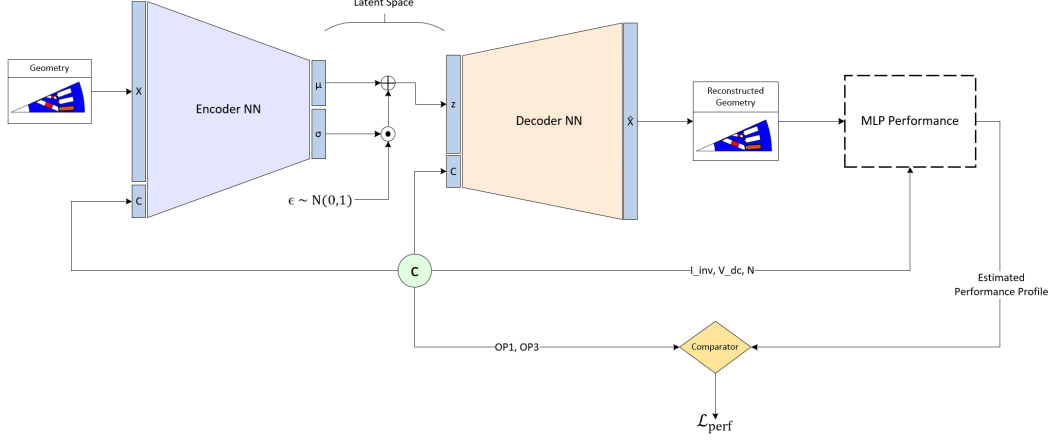


FIGURE 5.3: Training scheme of the CVAE with extra performance regulariser.

5.3.1 Regularisation Term

The performance-regularised CVAE augments the standard objective with an operating-point compliance penalty driven by the frozen MLP surrogate of Chapter 4. The surrogate is kept frozen: its parameters are never updated during CVAE training. Gradients of the penalty nonetheless propagate back through the decoder via the differentiable data path: at every training step the decoder output $\hat{\mathbf{x}}$ is clamped to $[0, 1]$, denormalised back to physical geometry, and passed to the surrogate. The surrogate predicts the PCA coefficients of the torque–speed curve, which are reconstructed on an 80-point speed grid from 250 to 20,000 rpm (in 250 rpm steps), yielding a differentiable proxy $\hat{T}(\omega)$ for the torque profile.

The two operating points encoded in $\mathbf{C}$ are penalised through a squared positive-shortfall:

$$\mathcal{L}_{\text{perf}} = \underbrace{\frac{1}{|\mathcal{M}_{\text{CT}}|} \sum_{\omega \in \mathcal{M}_{\text{CT}}} \left[\frac{\max(0, T_{\text{OP1}} - \hat{T}(\omega))}{|T_{\text{OP1}}| + 1} \right]^2}_{\text{OP1, constant-torque region}} + 10 \cdot \underbrace{\left[\frac{\max(0, T_{\text{OP3}} - \hat{T}(\omega_{\text{OP3}}))}{|T_{\text{OP3}}| + 1} \right]^2}_{\text{OP3, field-weakening point}}, \quad (5.5)$$

where $\mathcal{M}_{\text{CT}} = \{\omega \leq \omega_{\text{OP1}}\}$ is the region of the speed grid that corresponds to the constant-torque region. The OP3 contribution is scaled by 10 to compensate for the OP1 term being averaged over multiple speed points while OP3 is evaluated at a single point. Only shortfalls (predicted torque below the requested target) are penalised, so over-performing geometries are not discouraged. This additional penalty steers the decoder towards geometries that meet the requested operating points more often.

Modified Objective

The full training loss adds the compliance penalty to the standard ELBO:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta D_{\text{KL}}^{\text{FB}} + \lambda_{\text{perf}} \mathcal{L}_{\text{perf}}, \quad (5.6)$$

with the β -annealing of §5.2 unchanged. A two-stage schedule is used for λ_{perf} : it is held at 0 for the first 100 epochs (stage *S1*, a pure CVAE warm-up), linearly ramped from 0 to $\lambda_{\text{max}} = 1$ over epochs 100–150 (stage *S2*), and held constant thereafter. The two-stage schedule lets the latent space organise itself before the performance signal is introduced, which proved necessary to avoid early-training instabilities; gradient clipping is also tightened from 1.0 to 0.5 for the same reason. All other training settings match the standard CVAE.

5.4 Inference Procedure

Given a condition vector $\mathbf{C}$, a candidate motor geometry is generated by sampling $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and decoding:

$$\hat{\mathbf{x}} = f_{\theta}(\mathbf{z}, \mathbf{C}). \quad (5.7)$$

The sigmoid output $\hat{\mathbf{x}} \in [0, 1]^{d_x}$ is mapped back to physical geometry. Any number of independent candidates can be drawn for the same $\mathbf{C}$ by repeating the sampling step. Generated candidates are subsequently evaluated either through the frozen MLP surrogate (used in the case study of §5.5.3) or through the ZONDA toolchain.

5.5 Evaluation

This section evaluates both CVAE variants along two complementary axes: model-internal quality (Sections 5.5.1–5.5.2) and end-to-end usefulness as a geometry generator (§5.5.3).

5.5.1 Reconstruction Quality

Reconstruction quality measures how faithfully the encoder–decoder pair restores a held-out test-set geometry (the 15 % test partition of the train/validation/test split of Chapter 3) from its latent encoding. For every test sample, the encoder mean $\mu(\mathbf{x}, \mathbf{C})$ is decoded without reparameterisation noise, the sigmoid output is mapped back to physical geometry by inverting the min–max scaling of §5.1.3, and the per-feature residual ($\hat{G}_j - G_j$) is recorded in physical units.

Four metrics are reported per feature: MAE, RMSE, R^2 and MAPE. Because the 17 geometric parameters live on very different physical scales, a single MAE or RMSE in physical units would simply track whichever parameter has the largest absolute range. The aggregate values in Table 5.1 therefore divide each per-feature MAE and RMSE by the data range $G_{\max,j} - G_{\min,j}$ before averaging; R^2 and MAPE are dimensionless by construction and are averaged unweighted across the 17 features.

TABLE 5.1: Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for both CVAE variants.

Variant	MAE [% of range]	RMSE [% of range]	R^2	MAPE [%]
Standard CVAE	6.13	7.95	0.923	5.67
Regularised CVAE	6.19	8.07	0.921	5.63

Both variants reconstruct a typical test-set geometry to within roughly 6 % of feature range and explain about 92 % of the per-feature variance. In aggregate the regularised variant essentially matches the standard one, so the performance penalty reshapes the latent space without degrading average reconstruction fidelity.

The per-feature breakdown in Appendix C, §C.1 nuances this in two places. KGAMMA, KPQ1 and KPQ2 show inflated MAPE (12–16 %) due to near-zero targets. The regularised variant also reconstructs KMQ1 less accurately ($R^2 : 0.928 \rightarrow 0.875$), the only feature on which the two variants meaningfully diverge.

A second feature-level observation is that residuals widen systematically towards the lower and upper extremes of the normalised feature range, visible on the diagonal of Figure 5.4 and across the full grid in Appendix C, §C.1. The mechanism behind this and two alternative reconstruction objectives are discussed in Appendix C, §C.3.

Figure 5.4 shows parity plots for the three most condition-sensitive features identified in §3.3.1 (KAR, DSO, KMQ1); the per-feature breakdown over all 17 parameters and the full parity grid are reported in Appendix C, §C.1.

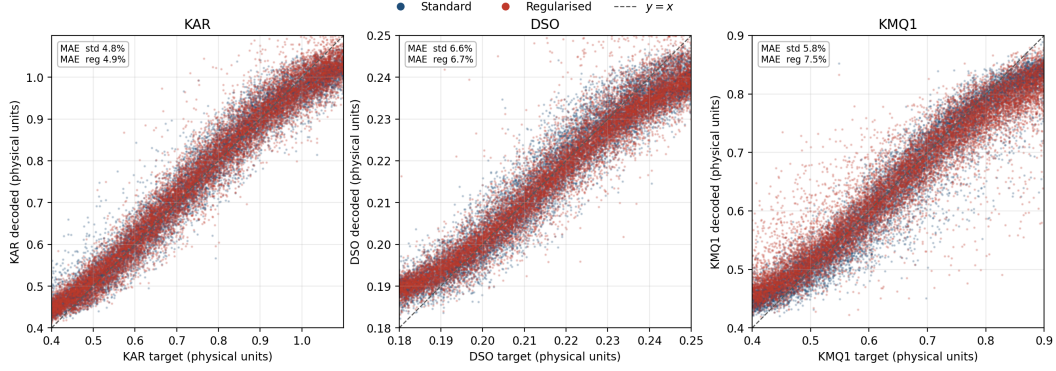


FIGURE 5.4: Parity plots of decoded vs. target geometry for the most condition-sensitive features (KAR, DSO, KMQ1) of §3.3.1. Standard and regularised CVAE are overlaid in distinct colours; the diagonal denotes perfect reconstruction.

5.5.2 Latent Space Utilisation and Interpretability

Beyond reconstruction fidelity, a useful latent space should make active use of its dimensions and assign each of them a recognisable design role. Two complementary views are reported below; the empirical marginal posteriors $q(z_k)$ vs. $\mathcal{N}(0, 1)$ for all 15 dimensions and both variants are additionally reported in Appendix C, §C.2.

Per-dimension KL

Figure 5.5 shows the average KL contribution $D_{KL,k}$ over the test set for each of the 15 latent dimensions, with the free-bits floor $\tau = 0.1$ nat overlaid. Dimensions whose averaged KL drops below τ are effectively unused; the count of collapsed dimensions is read off the figure directly.

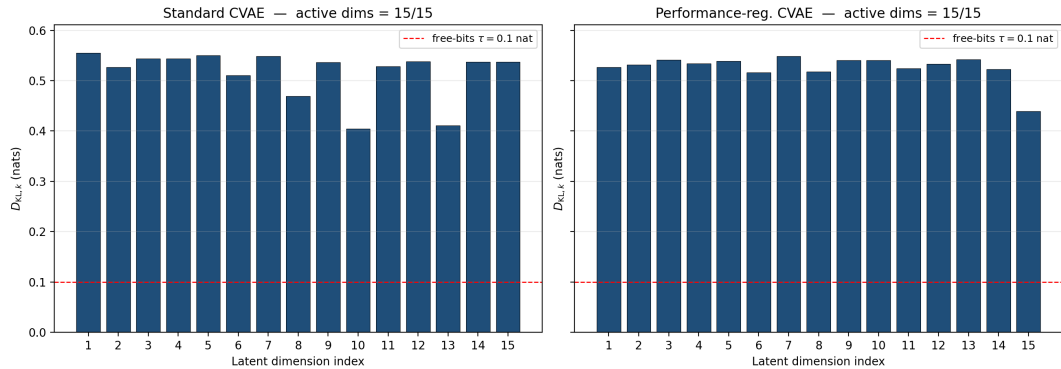


FIGURE 5.5: Per-dimension KL divergence on the test set for the standard (left) and regularised (right) CVAE.

Semantic meaning

Latent traversals (Figure 5.6) decode the variation of the top-3 most active latent dimensions while holding $\mathbf{C}$ and the remaining z_k fixed. Each varied dimension is swept from -2 to $+2$ in three steps; the resulting rotor cross-sections expose which design feature each dimension controls. The analogous sweeps for all 15 dimensions of both variants are reported in Appendix C, §C.2.

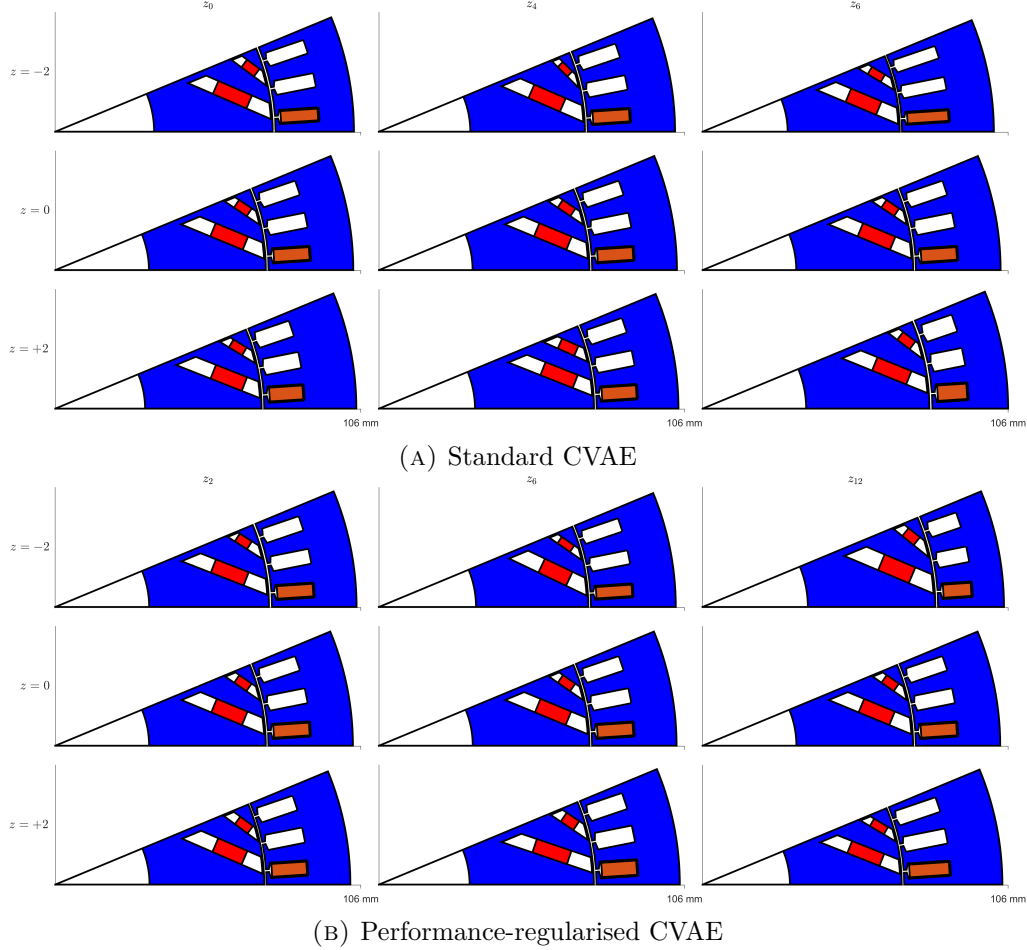


FIGURE 5.6: Latent traversals along the top-3 most active dimensions of the standard and regularised CVAE.

For each varied dimension, the figure shows one instance of the geometric diversity injected on top of the conditional baseline $\hat{\mathbf{G}}(\mathbf{z} = \mathbf{0}, \mathbf{C})$ for the chosen condition; the baseline itself moves with $\mathbf{C}$, so the same dimension can express slightly different design moves at a different operating point. Two examples illustrate the kind of structure recovered. In the standard variant, z_4 rotates the upper magnet, with its inclination angle shifting consistently across the sweep. In the regularised variant, z_{12} jointly narrows and lengthens the copper slots from $z = -2$ to $z = +2$, while the yoke thickens visibly and the rotor shrinks accordingly.

5.5.3 Case Study: 8 Industrial Conditions

The end-to-end performance of both models is assessed on 8 representative operating conditions provided by Punch Powertrain, drawn from outside the dataset. For each condition, candidate geometries are drawn from the prior (§5.4) without any selection or filtering, so the reported compliance and diversity reflect the unbiased behaviour of the generator. The aggregate metrics in this section are computed on $n = 1000$ samples per condition; the torque–speed plots of Figures 5.7 and C.14 use $n = 100$ samples per condition, to keep the curve bundles readable. All candidates are evaluated with the MLP surrogate of Chapter 4; a candidate is declared compliant when its surrogate-predicted torque satisfies equation (3.4) at both OP1 and OP3. Given the surrogate fidelity established in Chapter 4, no FEM evaluation is performed at this stage. Table 5.2 lists the 8 conditions; Table 5.3 reports per-condition compliance and range fraction for both variants.

TABLE 5.2: The 8 industrial conditions used for the case study, drawn from outside the dataset.

ID	ω_{OP1} [rpm]	T_{OP1} [Nm]	ω_{OP3} [rpm]	T_{OP3} [Nm]	I_{max} [A]	U_{DC} [V]	N
1	3672.81	790	15000	159.16	800	520	2
2	4743.85	310	14231.53	103.33	500	350	1
3	3846.25	720	13000	183.64	750	540	2
4	5729.58	250	18000	67.38	500	350	3
5	5601.04	520	18000	121.36	450	670	2
6	3370.34	850	12000	238.73	750	520	4
7	5835.68	180	16000	65.65	420	320	3
8	11317.68	135	20000	76.40	500	320	2

TABLE 5.3: Compliance and range fraction per industrial condition (Conditions 1–8 of Table 5.2), evaluated through the MLP surrogate on 1000 samples per condition.

ID	Compliance [%]		Range Fraction [%]	
	Standard	Regularised	Standard	Regularised
1	39.2	99.1	91.2	88.9
2	16.4	53.3	88.5	86.7
3	41.2	99.7	91.8	89.9
4	67.9	82.3	86.6	87.5
5	42.1	74.0	84.9	90.0
6	0.0	1.3	77.0	71.1
7	95.0	97.1	87.2	87.4
8	91.0	98.4	82.4	82.8
Average	49.1	75.7	86.2	85.5

Compliance

Aggregated over the 8 conditions, the regularised CVAE reaches a mean compliance of 75.7% versus 49.1% for the standard variant, while the mean range fraction barely moves (85.5% vs. 86.2%; see below). Qualitatively, the gain is concentrated at intermediate-difficulty conditions: Conditions 7 and 8 are already near-saturated for the standard variant and gain little, whereas Conditions 1 and 3 jump from roughly 40% to near-complete compliance under regularisation. Condition 6 (the highest-torque target) remains the hardest for both variants and is reached by only about 1% of regularised samples through unfiltered prior sampling. Figure 5.7 shows the underlying torque–speed envelopes for the first four conditions; the remaining 4 are reported in Appendix C, §C.4.

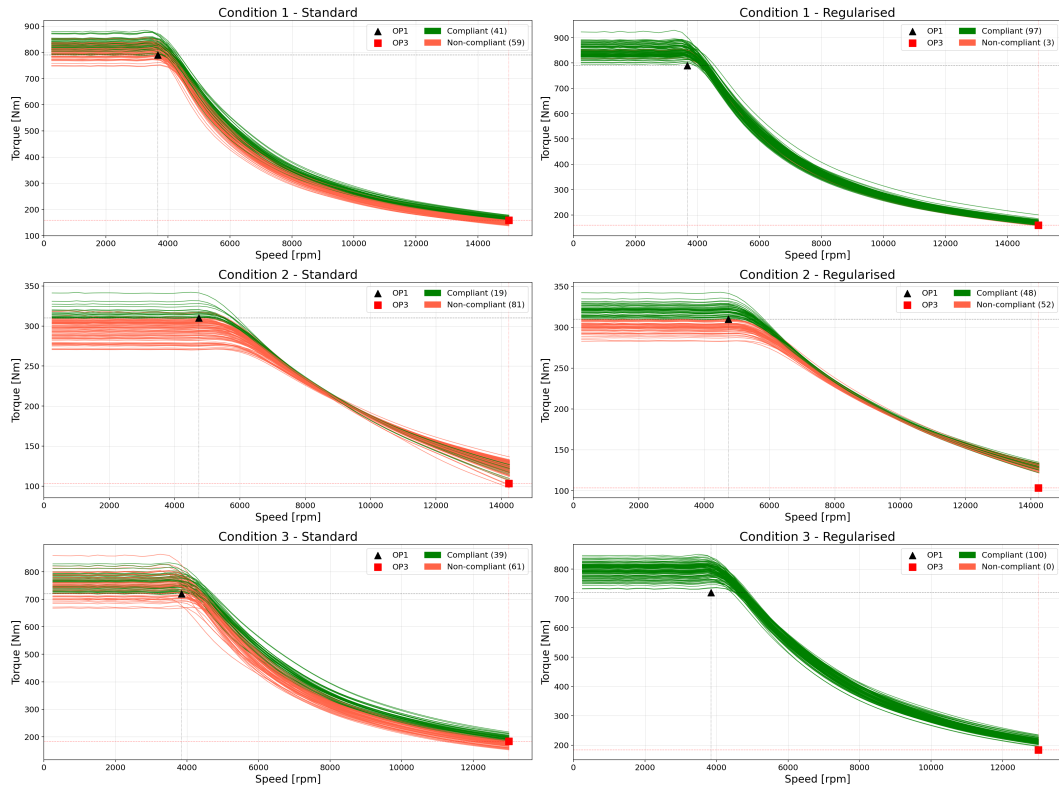


FIGURE 5.7: Torque–speed curves of 100 candidate geometries generated for 4 of the 8 industrial conditions, evaluated through the MLP surrogate. Compliant samples are coloured green, non-compliant samples red; target operating points OP1 and OP3 are marked.

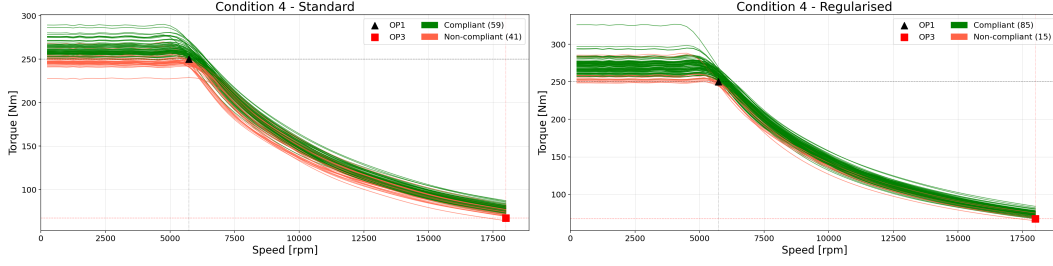


FIGURE 5.7: (continued).

Diversity

The diversity of the generated candidates is examined along three orthogonal axes:

- *Range fraction*: how wide the cloud of generated geometries is per dimension (range fraction, Table 5.3).
- *Distribution shape*: how the mass is laid out inside that range (per-dimension histogram with KDE overlay, Figure 5.8).
- *Joint correlations*: whether the generator preserves the unconditional independence between the design parameters or collapses onto a sub-manifold (aggregate Spearman matrix, Figure 5.9).

All three are computed on the full $n = 1000$ sample per condition, irrespective of compliance, so that the compliance and diversity numbers in Table 5.3 can be read against each other without effective-sample-size corrections.

Range fraction. The range fraction is defined per geometric parameter j as the ratio of the min–max span of the 1000 generated samples to the min–max span of that parameter on the full training set, and aggregated as the unweighted mean over the 17 parameters. Table 5.3 shows that both variants exercise on average roughly 86% of the per-feature design range; the regularised variant gives up almost nothing on range coverage for its compliance gain. The per-feature breakdown is reported in Appendix C, §C.4.

Distribution shape. Figure 5.8 shows, for the first two conditions, a 17-panel grid with one geometric parameter per panel. Each panel overlays the histogram and a kernel-density estimate of the 1000 generated samples for both variants. The remaining 6 conditions are reported in Appendix C, §C.4. Across most panels the two variants produce broad and similarly shaped marginal distributions; visible differences are localised and modest.

One important aspect should be noted about these graphs: the range fraction is normalised against the unconditional dataset span, so where a condition pinches the feasible region tightly the generator can produce a narrow marginal intrinsically

5. CONDITIONAL VARIATIONAL AUTOENCODER

rather than from collapse. A narrower distribution should therefore not be read automatically as a loss of diversity.

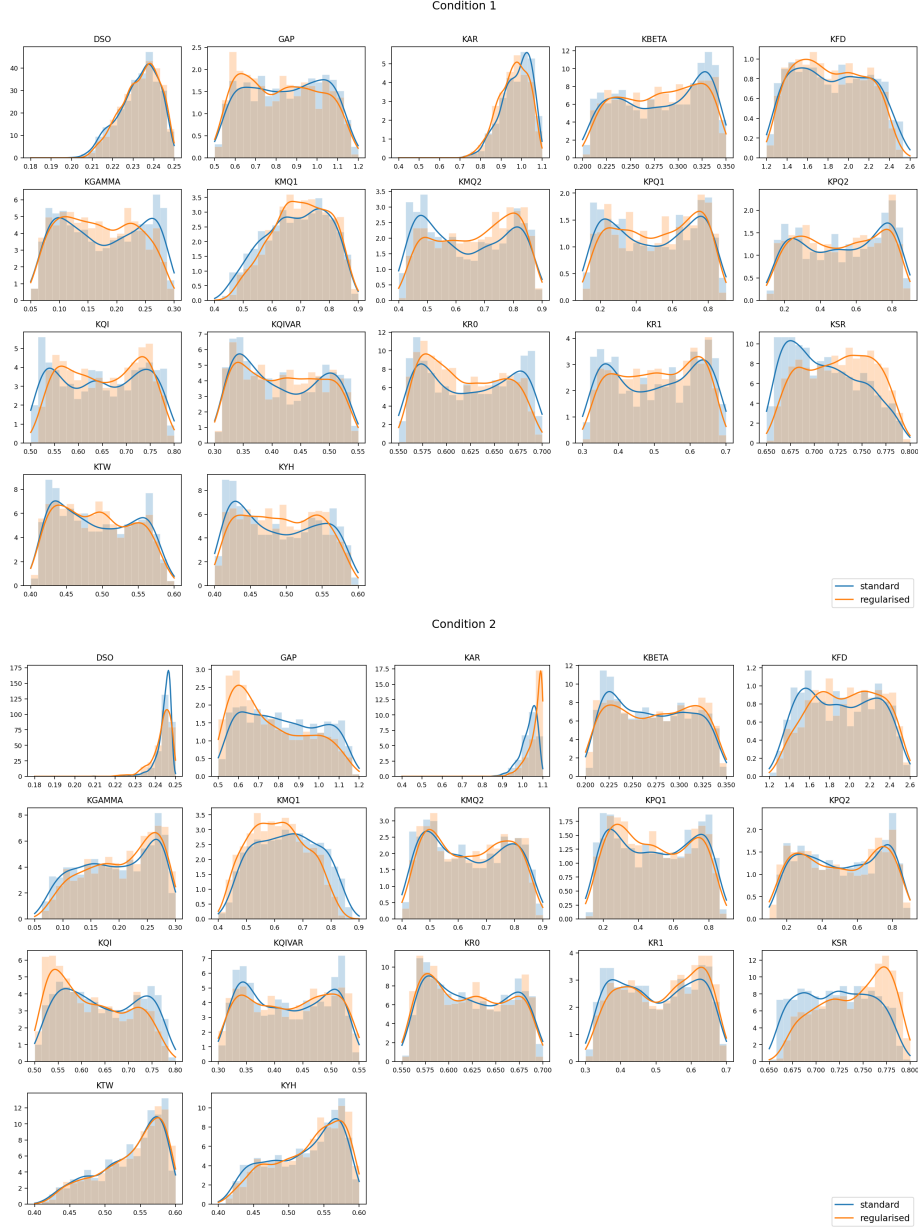


FIGURE 5.8: Marginal distributions of the 1000 generated samples for the first two industrial conditions, both model variants.

As a concrete example, DSO and KAR together describe the physical size of the machine, and the flux ϕ that this geometry can carry is limited at OP1 by the back-EMF relation $U_{back} = N \phi \omega$. Since U_{max} , N and ω_{OP1} are all fixed within a given condition, the admissible flux is effectively bounded by the ratio $U_{max}/(N \omega_{OP1})$,

and because that flux is provided by the iron in the machine, DSO and KAR are jointly constrained by this ratio rather than by $U_{\max}$ on its own. Shifts in their marginals across conditions should therefore be read against this combined limit.

Joint correlations. Figure 5.9 shows a single 17×17 Spearman correlation matrix, averaged element-wise across the 8 conditions. The lower triangle corresponds to the standard variant, the upper triangle to the regularised one. The dataset was sampled to be orthogonal in the design parameters (Chapter 3), so its unconditional Spearman matrix is approximately zero off-diagonal; the entries seen here are therefore correlations the generator itself has introduced. Most cells stay near zero ($|r| \lesssim 0.2$). The strongest cells, mainly in the DSO–KAR, KFD–KMQ1 and KAR–KTH pairs, reach magnitudes around 0.3–0.5 in either triangle; some such structure is unavoidable when 17 design parameters are compressed through a 15-dimensional latent and decoded back to 17. Spearman is used here for consistency with the data-analysis chapter and for robustness to monotone non-linear trade-offs and outliers.

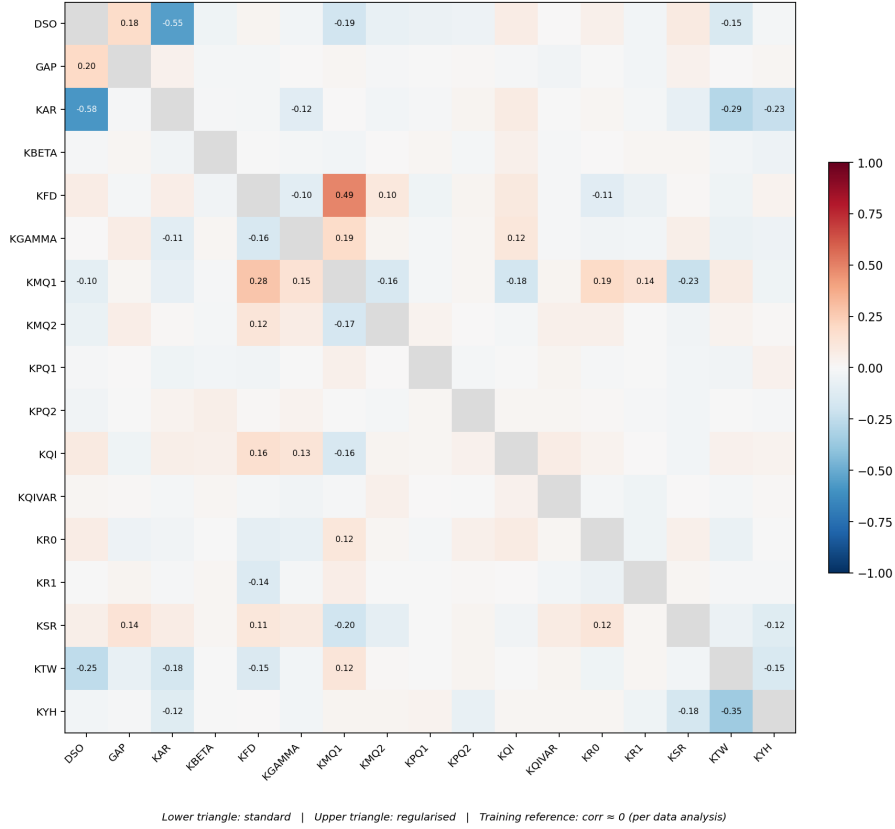


FIGURE 5.9: Aggregate Spearman split-correlation matrix of the predicted geometric features, averaged over the 8 industrial conditions. Lower triangle: standard CVAE; upper triangle: performance-regularised CVAE.

A positive cell at (i, j) indicates that within a single condition, samples with a high value for parameter i also tend to have a high value for parameter j , so that the two marginals in Figure 5.8 move in sync at the sample level; a negative cell indicates the opposite. The matrix is in that sense complementary to Figure 5.8: the marginals describe what each parameter does on its own, whereas the matrix describes how parameters move together within a sample.

As a concrete example of how the within-condition view differs from the across-condition view, consider the KFD–KMQ1 pair. From one condition to the next the two marginals tend to shift in opposite directions, most clearly visible on the appendix conditions (§C.4) where $\mathbf{C}$ pinches the feasible range of both parameters more tightly. This follows directly from the dataset: in Figure B.1 both parameters are mainly driven by ω_{OP3} but with opposite signs (+0.16 vs. -0.23), and the CVAE reproduces that conditional shift. The positive KFD–KMQ1 cell in Figure 5.9 captures something else: the per-condition Spearman averaged over the 8 conditions, i.e. the coupling the decoder introduces within each cloud at a fixed $\mathbf{C}$.

Summary

Both CVAE variants reconstruct the test set to within $\approx 6\%$ of feature range ($R^2 \approx 0.92$), so the performance penalty reshapes the latent space without measurably degrading fidelity. On the 8 industrial conditions, the regularisation lifts mean compliance from 49.1% to 75.7% while the average range fraction ($\approx 86\%$), the marginal distributions and the aggregate correlation structure are essentially unchanged. The gain concentrates at intermediate-difficulty conditions; Conditions 7 and 8 are already near-saturated under the standard variant, and Condition 6 remains hard for both ($\approx 1\%$ compliance).

Three limitations of unfiltered prior sampling motivate the next chapter: roughly a quarter of regularised samples still miss OP1 or OP3, and the generator optimises no design cost (e.g. machine volume) beyond the conditioning itself. Moreover, so far we have omitted the structural analysis of the sample, described in Appendix A.2. Chapter 6 addresses these limitations by holding the decoder fixed and refining $\mathbf{z}$ by gradient descent under explicit compliance and structural penalties.

Chapter 6

Latent Space Optimisation

6.1 Problem Formulation

The inference procedure of §5.4 generates valid motor geometries conditioned on a target operating point $\mathbf{C}$, but random sampling gives no control over secondary design objectives such as mass, material cost, volume, or efficiency. This chapter closes that gap by adding a second inference stage in which the latent variable $\mathbf{z}$ is refined by gradient descent towards a chosen secondary objective. In the present work that objective is minimal machine volume.

The optimisation variable is $\mathbf{z} \in \mathbb{R}^{15}$. The decoder $f_\theta(\cdot, \mathbf{C})$ has frozen weights and is differentiable in $\mathbf{z}$, so any differentiable expression appended to its output can be back-propagated to $\mathbf{z}$ via the chain rule. Compared to optimising directly in the 17-dimensional geometry space $\mathbf{G}$, this latent-space formulation is attractive for two reasons. First, the latent space is lower-dimensional and concentrated around $\mathcal{N}(\mathbf{0}, \mathbf{I})$ by construction, which bounds the search region and is expected to facilitate gradient-based optimisation. Second, performance compliance is implicit by construction: the condition vector $\mathbf{C}$ supplied to the decoder already encodes the requested performance (through OP1 and OP3) and the electrical budget (U_{max}, I_{max}). The decoder has therefore been trained to map $(\mathbf{z}, \mathbf{C})$ to geometries that approximately meet these targets, so a random $\mathbf{z}_0$ already decodes to a near-compliant design. The optimiser only refines $\mathbf{z}$ to trade off the secondary objective (volume) against any residual constraint violations, rather than searching the full 17-dimensional hypercube where most points fail the performance constraints by a wide margin.

The constrained problem is

$$\min_{\mathbf{z} \in \mathbb{R}^{15}} \frac{v(\hat{\mathbf{x}})}{V_{\text{ref}}} \quad \text{s.t.} \quad \begin{cases} \hat{T}_i(\hat{\mathbf{x}}, N, U_{max}, I_{max}) \geq T_{\text{OP}_i}, & i \in \{1, 3\}, \\ \text{KFD}(\hat{\mathbf{x}}) \cdot 8000 \geq \omega_{\text{OP}_3}, \end{cases} \quad (6.1)$$

$$\text{with } \hat{\mathbf{x}} = f_\theta(\mathbf{z}, \mathbf{C}),$$

where $v(\hat{\mathbf{x}})$ is the volume of the decoded machine geometry $\hat{\mathbf{x}}$, evaluated analytically as

$$v = (\text{DSO} \cdot \text{KAR}) \cdot \left(\frac{\text{DSO}}{2} \right)^2 \cdot \pi, \quad (6.2)$$

with DSO the stator outer diameter and KAR the length-to-diameter aspect ratio (both contained in $\hat{\mathbf{x}}$). V_{ref} is the mean volume of the dataset, which rescales the objective to order one so that all loss terms remain comparable in magnitude. The performance constraint requires the torque predicted by the MLP surrogate of Chapter 4, $\hat{T}_i$, to reach at least the requested torque at the two operating points OP1 and OP3 contained in $\mathbf{C}$.

The structural constraint requires the mechanical critical speed $\text{KFD} \cdot 8000 \text{ rpm}$ to exceed the highest requested operating speed ω_{OP3} . This product defines a hard mechanical limit on the rotor's maximum admissible rotational speed: above $\text{KFD} \cdot 8000 \text{ rpm}$ the centrifugal stresses exceed the rotor's material strength and the design loses its structural integrity.

6.2 Gradient-Based Method

The decoder f_θ is a neural network with frozen weights, the MLP surrogate of Chapter 4 is likewise frozen and differentiable, the volume v is an analytical function of the decoded geometry while KFD is one of its 17 components. The full chain from $\mathbf{z}$ to the scalar loss is therefore differentiable end-to-end and back-propagation yields a usable gradient $\partial \mathcal{L} / \partial \mathbf{z}$ at every iteration:

$$\mathbf{z} \longrightarrow f_\theta(\mathbf{z}, \mathbf{C}) = \hat{\mathbf{x}} \longrightarrow (v(\hat{\mathbf{x}}), \text{KFD}, \hat{T}_1, \hat{T}_3) \longrightarrow \mathcal{L}(\mathbf{z}). \quad (6.3)$$

Three mechanisms are added to the loss to keep the trajectory inside the trustworthy region of latent space:

$$\begin{aligned} \mathcal{L}(\mathbf{z}) = & \frac{v(f_\theta(\mathbf{z}, \mathbf{C}))}{V_{\text{ref}}} + \alpha \|\mathbf{z} - \mathbf{z}_0\|^2 \\ & + \beta \sum_{i \in \{1,3\}} \max\left(0, 1 - \frac{\hat{T}_i}{T_{\text{OP}_i}}\right) + \gamma \max\left(0, 1 - \frac{\text{KFD} \cdot 8000}{\omega_{\text{OP3}}}\right). \end{aligned} \quad (6.4)$$

The four terms play distinct roles:

- $\frac{v(f_\theta(\mathbf{z}, \mathbf{C}))}{V_{\text{ref}}}$: the normalised secondary objective evaluated on the decoded geometry. This is the quantity actually being minimised; the other three terms only constrain the search.
- $\alpha \|\mathbf{z} - \mathbf{z}_0\|^2$: a trust-region regulariser that anchors the optimisation around the initial latent code $\mathbf{z}_0$. Penalising the squared distance keeps the trajectory close to the region of the prior in which the decoder was trained and prevents large jumps into out-of-distribution latent regions. The weight α sets the size of the trust region.

- $\beta \sum_i \max(0, 1 - \frac{\hat{T}_i}{T_{OP_i}})$: a compliance penalty driven by the frozen MLP surrogate at the two operating points. The $\max(0, \cdot)$ activates the term only when the predicted torque falls below the target, so compliant candidates incur no penalty; the weight β controls how strictly the conditioning is preserved during optimisation.
- $\gamma \max(0, 1 - \frac{\text{KFD} \cdot 8000}{\omega_{OP3}})$: a structural penalty driven by the KFD component of the decoded geometry. It activates only when the rotor's critical speed falls below the highest requested operating speed and steers $\mathbf{z}$ back towards mechanically stable geometries.

Optimiser, initialisation, and multi-start

Updates of $\mathbf{z}$ use Adam with learning rate $\eta = 0.05$ for $n_{\text{iter}} = 500$ iterations. To hedge against poor local optima the optimiser is restarted $N_{\text{starts}} = 25$ times from independent random initialisations

$$\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, T^2 \mathbf{I}) \quad (6.5)$$

The anchor $\mathbf{z}_0$ appearing in the trust-region term is taken to be the start of that particular run. After all trajectories have converged, the *winner* is selected as the endpoint with the smallest volume that is fully compliant on both penalty terms; if no run ends compliant, the run with the lowest $\mathcal{L}$ is returned as a fallback and flagged as infeasible. Figure 6.1 summarises the gradient loop, and the hyperparameters used in this work are listed in Table 6.1.

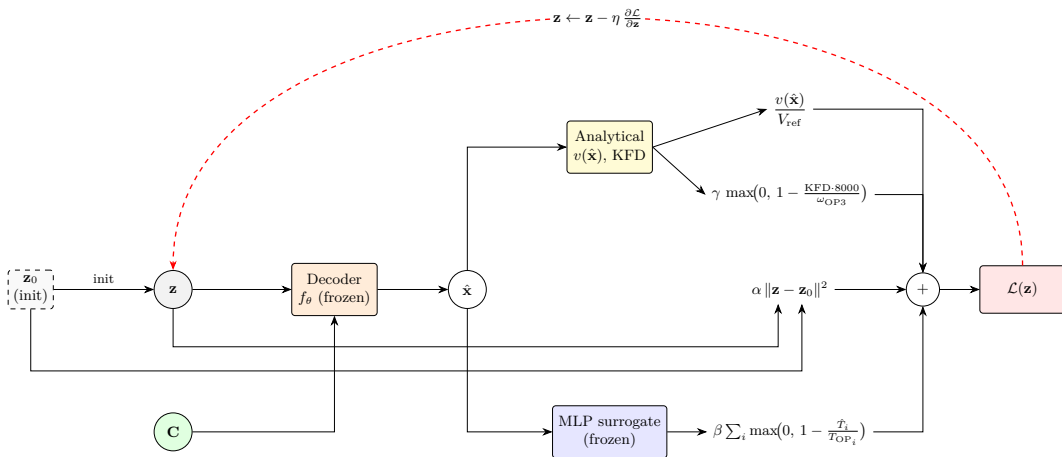


FIGURE 6.1: Gradient-based latent-space optimisation loop. The dashed red arrow indicates the gradient update on $\mathbf{z}$; all networks remain frozen.

TABLE 6.1: Hyperparameters of the latent-space optimiser.

Symbol	Meaning	Value
N_{starts}	independent Adam runs (multi-start)	25
n_{iter}	iterations per run	500
η	Adam learning rate	0.05
T	latent-init temperature, $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, T^2 \mathbf{I})$	1.5
α	trust-region weight	0.01
β	performance-violation weight	10
γ	structural-violation weight	10

6.3 Results

The latent-space optimiser of §6.2 is run for the eight industrial conditions of Table 5.2, each driven by both the standard and the regularised CVAE. Condition 1 is reported in full below; the corresponding figures for Conditions 2–8 are collected in Appendix D.

TABLE 6.2: Latent-space optimisation winners across the 8 industrial conditions of Table 5.2, for both CVAE variants.

ID	Model	V [L]	Length [mm]	DSO [mm]
1	Standard	6.80	211.9	202.1
	Regularised	6.94	206.5	206.8
2	Standard	8.35	230.0	215.1
	Regularised	8.58	233.8	216.2
3	Standard	6.32	209.2	196.1
	Regularised	6.65	204.5	203.5
4	Standard	2.19	78.0	188.9
	Regularised	2.32	81.0	191.0
5	Standard	7.00	211.9	205.1
	Regularised	7.37	212.3	210.3
6	Standard	5.30	109.5	248.4
	Regularised	5.41	114.6	245.1
7	Standard	2.03	75.7	184.6
	Regularised	2.11	78.2	185.5
8	Standard	2.01	76.3	183.0
	Regularised	2.06	77.8	183.5

Table 6.2 summarises the winning geometries selected by the multi-start rule of §6.2. Both CVAE variants are able to produce compliant winners for all eight conditions.

Notably, the standard variant consistently arrives at a slightly smaller machine volume than the regularised variant.

The four convergence panels of Figure 6.2 show that the total loss settles to a stable plateau within roughly 100 iterations for all 25 multi-start runs, indicating that the search has not been ended in a wildly different attractor across restarts. The performance-violation and structural-violation terms are driven to zero quickly: for the regularised variant they are essentially flat at zero from the start, while for the standard variant $\text{viol}_{\text{perf}}$ departs from a higher initial value but reaches zero within the first ~ 50 iterations.

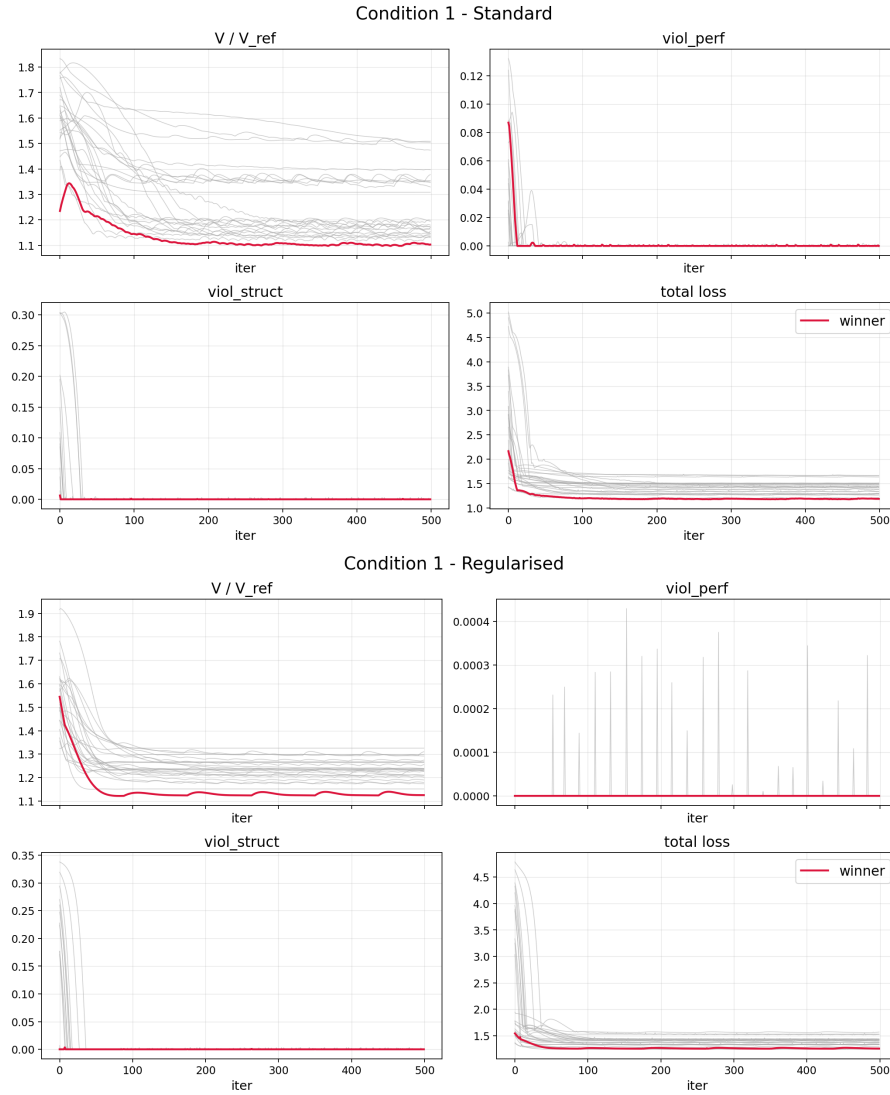


FIGURE 6.2: Convergence curves for the 25 multi-start runs of Condition 1; the winner is highlighted.

Figure 6.3 compares the 25 winning candidates of the multi-start (green vertical lines mark a compliant winner-per-start, red marks a non-compliant winner-per-start, and the thick red line is the overall winner returned by the selection rule) against 100 random CVAE samples drawn from the prior (grey histogram: all samples; blue histogram: the subset that satisfies both the performance and structural requirements). The overall winner lies below the minimum of the random-sample baseline for both CVAE variants, confirming that the gradient step delivers a genuine volume reduction rather than merely selecting the smallest random sample.

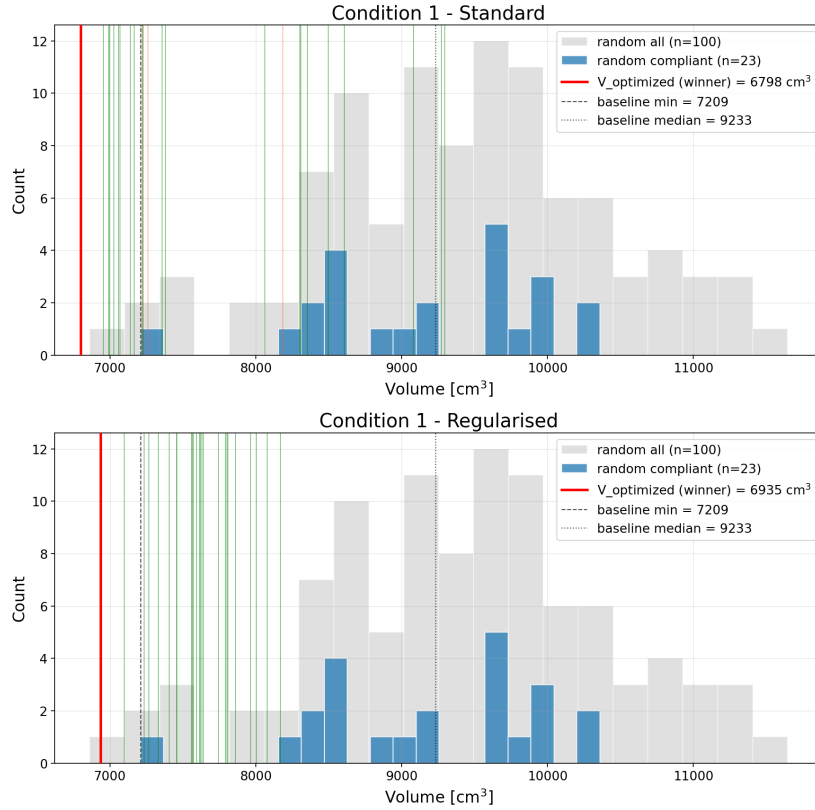


FIGURE 6.3: Volume comparison for Condition 1: the 25 multi-start endpoints (green; winner in red) against 100 random CVAE samples (grey: all, blue: compliant). Standard (top) and regularised (bottom) CVAE.

Figure 6.4 validates the optimiser’s winner evaluated through the MLP against the FEM ground truth of the decoded geometry. The MLP-predicted torque-speed curve tracks the FEM curve closely along the full speed range, and the curve passes through the requested torques at OP1 and OP3. The surrogate-driven optimisation therefore transfers to the true performance map for Condition 1.

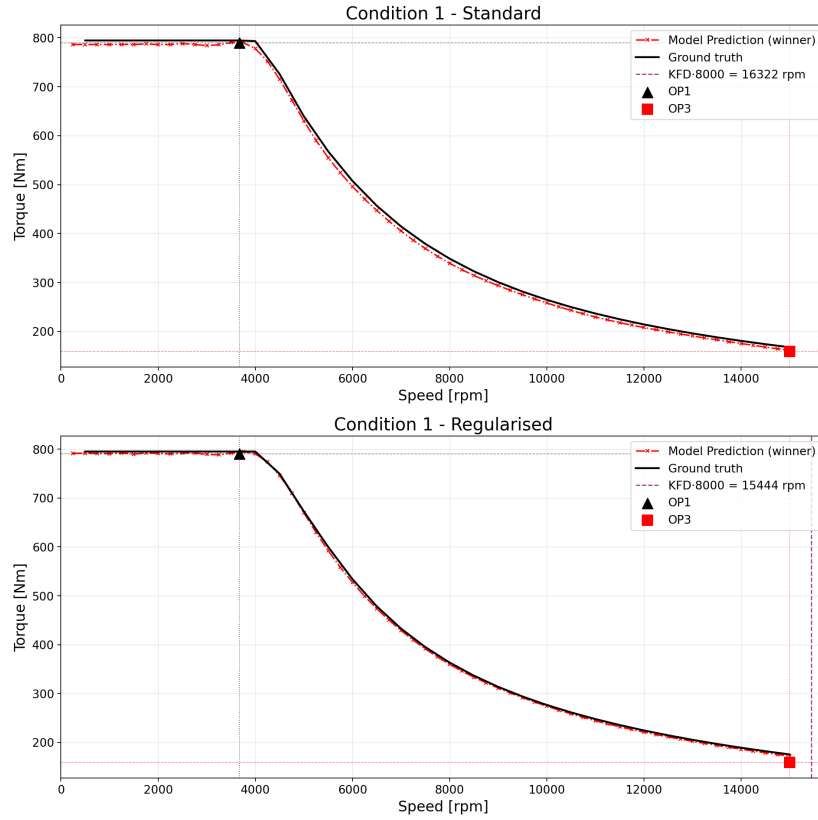


FIGURE 6.4: Validation of the winner for Condition 1: MLP-predicted torque-speed curve versus FEM ground truth, with OP1 and OP3 marked. Standard (top) and regularised (bottom) CVAE.

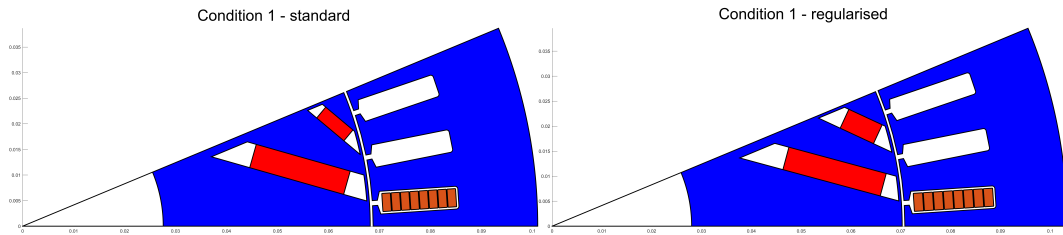


FIGURE 6.5: One-pole slice of the winning geometry for Condition 1. Standard (left) and Regularised (right) CVAE.

Figure 6.5 shows the one-pole slice of the Condition 1 winners for both CVAE variants. The most visible difference between both geometries for this particular condition lies in the magnet sizing and orientation: the upper magnet adopts a different angle and thickness between the two variants, the lower magnet is slightly shifted, and the standard variant settles on a slightly smaller rotor diameter than the regularised one. Stator slot geometry and air-gap thickness are nearly indistinguishable between the two.

The corresponding loss curves, baseline comparison, FEM validation, and one-pole drawings for Conditions 2–8 are collected in Appendix D. The qualitative behaviour observed for Condition 1 carries over to those conditions, with the one minor exception of Condition 4, where a small MLP/FEM discrepancy is noted in the appendix.

6.4 Discussion

The results of §6.3 support a few clear conclusions. The gradient loop produces a compliant winner for every one of the eight industrial conditions and for both CVAE variants (Table 6.2), so the method is not tied to a single operating point. Convergence is fast and consistent: the 25 restarts settle to a stable plateau within roughly 100 iterations and end up close to each other (Figure 6.2), which suggests that the optimisation lives in a well-behaved region of the latent space rather than jumping between very different attractors. A complete optimisation for one condition (25 restarts of 500 iterations each) takes about 45s on a single CPU core, which is negligible compared to a single KPI evaluation through the ZONDA toolchain (on the order of 20 min). The winners also lie below the smallest volume found in 100 random prior samples (Figure 6.3), so the improvement is real and not just a lucky draw. Finally, the optimum found through the MLP transfers to the FEM ground truth: the validated torque–speed curve passes through both OP1 and OP3 (Figure 6.4), which means the surrogate is accurate enough to drive the search, with FEM only needed for the final check.

The standard variant consistently ends up at a slightly smaller volume than the regularised one. This fits the picture from Chapter 5: the performance penalty pushes the regularised decoder towards more conservative geometries that are already close to compliance, which leaves less room for the latent search to explore towards lower volume. Both variants reach feasible designs, so the choice between them is a trade-off between higher compliance at sampling time and a slightly smaller optimum at inference time.

Chapter 7

Conclusion

7.1 Summary

The current ZONDA-based design workflow at Punch Powertrain wraps a FEM-based KPI evaluator in an evolutionary loop, and a single customer case takes several days on a multi-core node because each candidate requires on the order of 20 min of single-core CPU time and thousands of candidates are issued before convergence [27, 8]. This thesis attacked the same problem from the inverse direction: given a target peak performance profile together with the electrical limits and the winding ratio, can a generative model directly sample a diverse set of compliant double-V IPMSM geometries, and can those samples subsequently be refined towards a secondary objective?

The proposed framework, trained on a single dataset of 85 384 physically valid LHS samples of the 17 free geometric parameters (Chapter 3), combines three components: a residual MLP surrogate that maps geometry and electrical limits to the 80-point peak torque–speed curve through a 19-component PCA bottleneck (Chapter 4); a CVAE that learns $p(\mathbf{G} \mid \mathbf{C})$ in two variants, a standard β -ELBO baseline and a performance-regularised variant whose loss adds a differentiable compliance penalty driven by the frozen surrogate (Chapter 5); and a gradient-based latent-space optimisation stage that refines a sampled candidate towards minimal volume under soft compliance and structural-integrity penalties driven by the same surrogate, with a trust-region anchor in latent space (Chapter 6). Relative to the two preceding theses in this series [27, 8], which accelerated the forward problem by surrogating the flux-map FEM step, this work attacks the inverse problem directly.

7.2 Main Findings

Surrogate MLP (Chapter 4)

On the held-out test set, the surrogate reaches a mean per-sample relative L_2 error of 0.62 % on the denormalised torque curve (median 0.41 %, 95th-percentile 2.01 %, Table 4.1) at sub-millisecond inference cost, well within the 1 % fidelity criterion adopted in §4.4 and sufficient to act as a differentiable, in-loop substitute for FEM in both downstream stages.

Standard CVAE (Chapter 5, §5.2)

The baseline reconstructs a typical test-set geometry to within ≈ 6 % of feature range ($R^2 \approx 0.92$, Table 5.1). On the eight industrial conditions of Table 5.2, unfiltered prior sampling yields a mean compliance of 49.1 % at a mean per-feature range fraction of 86.2 % (Table 5.3), with marginals and aggregate Spearman structure indicating no collapse onto a sub-manifold of the design space.

Performance-regularised CVAE (Chapter 5, §5.3)

Augmenting the ELBO with the differentiable compliance penalty lifts the mean compliance from 49.1 % to 75.7 % while the range fraction barely moves (86.2 % $\rightarrow$ 85.5 %) and reconstruction accuracy is essentially unchanged. The gain concentrates at intermediate-difficulty conditions; Condition 6 remains hard for both variants (≈ 1 % compliance), which motivates the second-stage refinement.

Latent-space optimisation (Chapter 6)

The gradient search returns a compliant candidate geometry for every condition and for both CVAE variants (Table 6.2), with the 25 multi-start trajectories settling within roughly 100 iterations (Figure 6.2) and the winner lying below the smallest volume of 100 random prior samples (Figure 6.3). The standard variant arrives at slightly smaller volumes than the regularised one: the regularisation pushes the decoder towards conservative, near-compliant geometries and leaves less room for the latent search to trade compliance against volume. A FEM validation of the Condition 1 winner confirms that the surrogate-driven optimum transfers to the true performance map (Figure 6.4). A complete optimisation per condition takes about 45 s on a single CPU core, against the order-of-day cost of the current ZONDA loop.

The combined picture is that the generator and the optimiser act as a pair: the CVAE supplies the diversity and a near-compliant starting point, the latent search closes the residual gap and adds the secondary-objective leverage. The framework compresses what is currently a days-long search per customer case into a sub-minute generative step.

7.3 Limitations and Outlook

The scope of the framework is deliberately narrow, and each of the limitations below points to a natural direction for further work.

Demagnetisation is not modelled: The MLP predicts torque from the geometry and the electrical limits, but it has no notion of the magnetic field acting on the magnets themselves. During field weakening at high speed, the optimiser is free to use a large negative i_d to reach OP3, and the resulting opposing flux can, for a thin or unfavourably oriented magnet, exceed the irreversible demagnetisation threshold of the magnet material. The winner is therefore guaranteed to be torque-compliant and structurally safe up to $\text{KFD} \cdot 8000 \text{ rpm}$, but not guaranteed to keep its magnetisation across the full speed range. A surrogate predicting the minimum magnet operating point, plugged into the loss as a soft penalty mirroring the existing torque and KFD terms, would close this gap and is the highest-priority extension.

Thermal and loss behaviour are out of scope: Copper, iron, and magnet losses, and the temperature rise that follows from them, are not part of the loss during optimisation. Two geometries with the same volume and the same torque can have very different efficiencies, and the present scheme does not distinguish between them. An efficiency surrogate inserted into the second-stage loss as a soft penalty or as a Pareto axis is the natural multi-objective extension.

Surrogate trust: The latent search trusts the MLP throughout the 500×25 iterations per condition; the end-of-loop FEM validation catches MLP–FEM gaps although they are small (such as the gap on Condition 4, Appendix D).

Fixed materials and topology: The lamination (HILITE N020 MEAS), magnet (N42UH) and rotor topology (double-V) are held fixed. Generalising to a multi-topology setting, with the topology as a discrete component of $\mathbf{C}$ and a single decoder spanning multiple rotor families, is the most natural long-horizon direction for this line of work.

Appendices

Appendix A

Supplementary Theoretical Background

This appendix collects derivations and toolchain detail that support Chapters 1 and 2 but are not needed to follow their main thread.

A.1 Formal Multi-Objective Optimisation Formulation

Section 1.1.1 sketches the optimisation problem solved by the conventional Punch Powertrain workflow in qualitative terms. The corresponding formal statement is given below.

Let $\mathbf{G} \in \mathbb{R}^N$ denote the geometry vector of N free parameters. The multi-objective optimisation seeks to minimise the manufacturing cost, mass, and physical size while maximising the energy-conversion efficiency:

$$\min_{\mathbf{G}} [C_{\text{cost}}(\mathbf{G}), m(\mathbf{G}), V(\mathbf{G}), -\bar{\eta}(\mathbf{G})], \quad (\text{A.1})$$

where C_{cost} is the manufacturing cost, m the total mass, V a measure of size (typically the stator outer diameter or active volume), and $\bar{\eta}$ an aggregated efficiency.¹ Each quantity is evaluated through the DCP modules of §A.2.

The performance constraint requires the candidate to deliver at least the client-specified torque and power at every relevant speed:

$$T(\omega) \geq T_{\text{req}}(\omega), \quad P(\omega) \geq P_{\text{req}}(\omega), \quad \forall \omega \in [0, \omega_{\text{max}}]. \quad (\text{A.2})$$

¹Efficiency is not a property of the geometry alone but depends on the operating regime (T, ω) . Here $\bar{\eta}$ denotes a scalar aggregate, for example a weighted average over a representative drive cycle or the value at a nominal operating point.

A.2 The ZONDA Design Computation Platform in Detail

Section 1.1.2 introduces the Design Computation Platform (DCP) at a high level. The full per-module description and the internal sequencing of its three FEM-dependent modules are reproduced here [8].

The six KPI modules

1. **Structural Analysis.** Analytical evaluation of the mechanical robustness of the rotor laminations and magnet bridges. Verifies that the design withstands the centrifugal stress at peak speed.
2. **Cost Estimation.** Analytical estimate of manufacturing cost based on internal cost models, aggregating raw-material usage (magnets, copper, iron) and production effort (stamping, winding, assembly).
3. **Performance Evaluation.** 2D FEM-based evaluation of torque, power, and losses across the operating envelope. Also reports secondary electromagnetic effects such as demagnetisation margin and torque ripple.
4. **Efficiency Analysis.** Loss evaluation (copper, iron, magnet, mechanical) at a set of representative real-world operating points. The individual losses are weighted by their expected duty-cycle share to yield a single efficiency figure.
5. **Thermal Behavior.** FEM-based thermal simulation returning the time each component (winding, magnet, lamination) needs to reach its temperature limit, both at peak and continuous operating conditions. Directly determines the continuous performance profile.
6. **NVH.** Analytical estimate of radiated sound power as a function of frequency and rotational speed.

Of these six modules, three share a common computational backbone. The performance, efficiency, and thermal modules all depend on quantities that are not defined until the candidate’s actual performance profile is known. Only the *required* profile is supplied by the client; what the candidate can deliver must be recovered from its geometry. Figure A.1 shows the three-step pipeline by which the DCP transforms a geometry vector $\mathbf{G}$, subject to the electrical limits ($U_{\max}, I_{\max}$), into a performance profile (PP). The technical content of each step is developed in §2.1.3.

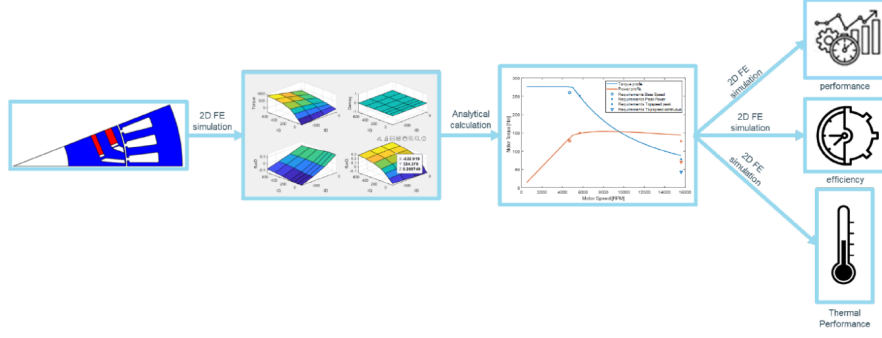


FIGURE A.1: Design Computation Platform process: a geometry $\mathbf{G}$ is mapped through a flux-map FEM step, a winding-ratio scaling, and an MTPA/field-weakening construction to a peak performance profile, subject to the electrical limits of the client.

Once the PP is available, the three PP-dependent KPIs follow. The Performance KPI is read directly from the PP. The Thermal KPI runs transient simulations across the operating range to derive the continuous envelope from the peak one. The Efficiency KPI combines that thermal analysis with the performance at the chosen speed range. The remaining three KPIs (Structural, Cost, NVH) bypass this pipeline and are evaluated analytically from $\mathbf{G}$ and the material selection directly.

A.3 The Arbitrary Reference Frame: Clarke and Park Transform

Section 2.1.2 introduces the dq frame used throughout the electromagnetic model of §2.1.2; the explicit transform from three-phase (a, b, c) quantities to the rotating $(q, d, 0)$ frame is reproduced here.

A generic three-phase quantity f_{abc} is transformed to the rotating frame via the Clarke and Park transforms [32]:

$$f_{qd0} = \mathbf{K} f_{abc}, \quad \mathbf{K} = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{4\pi}{3}\right) \\ \sin \theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{4\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad (\text{A.3})$$

where the gain $k = 2/3$ corresponds to the constant-amplitude convention, which preserves the amplitude of the original phase quantities and yields the most convenient form of the torque equation [26]. When the frame is locked to the rotor by setting $\omega = \omega_r$ and aligning the d -axis with the permanent-magnet flux vector, all inductances become constant and all voltages and currents become DC quantities in steady state. This specific choice is the *synchronous reference frame* used throughout the main text.

A.4 Winding Ratio: Technical Detail

Section 2.1.3 introduces the winding ratio N that bridges the flux map and the flux-linkage map. Two further points are treated here: the invariance of the underlying FE solution under rewinding, and the discreteness of N imposed by hairpin winding technology.

Poisson invariance of the flux map under rewinding

Section 2.1.3 states that a single FE flux map suffices to obtain the flux-linkage map for every winding ratio N . The justification rests on the structure of the underlying field equation. The 2D FE problem is governed by Poisson’s equation [44],

$$\nabla \cdot (\mu^{-1} \nabla A_z) = -J_z, \quad (\text{A.4})$$

where A_z is the z -component of the magnetic vector potential and J_z is its only source term. The source term is fixed by the imposed slot current density J , which does not depend on how the slot is wound: changing N redistributes the number of turns and the per-conductor current inside the slot, but leaves the total ampere-turns per unit cross-section unchanged. The field solution A_z , and hence the flux per slot ϕ_d , ϕ_q , therefore depend only on the geometry, the materials, and J , not on N .

Hairpin winding technology

The winding ratio N is not continuous: only a finite set of physical winding configurations exists. The target machine uses *hairpin* winding: rectangular, pre-formed U-shaped copper conductors inserted into the stator slots and welded at the opposite end. Compared with conventional random-wound stators, hairpin offers a high slot fill factor, better thermal conductivity, and is well suited to automated assembly, which is why it dominates modern automotive traction motors. Because the conductors are physical, pre-formed pieces, only a small discrete set of series/parallel arrangements is feasible: $N \in \{1, 2, 3, 4, 6, 8\}$ for the target machine. This thesis restricts attention to $N \in \{1, 2, 3, 4\}$, with the higher values omitted for tractability.

A.5 Neural-Network Training Techniques and Hyperparameter Detail

Section 2.4 names the hyperparameter and training-technique choices used for the surrogate of Chapter 4 and the CVAE of Chapter 5. This section expands on each of them.

Network depth and width

Depth (the number of hidden layers L) and width (the number of units per layer) jointly set how rich a mapping the MLP can represent. Wider layers add capacity within a fixed depth; additional layers compose representations and tend to be more parameter-efficient for highly non-linear mappings, at the cost of harder optimisation. Both too little capacity (underfitting) and too much (overfitting, longer training, less stable optimisation) hurt generalisation, and the right point on this trade-off is typically identified empirically on the validation set.

Activation functions

The element-wise nonlinearity f in (2.12) is what gives the MLP its expressive power; without it any depth of stacked affine layers would collapse into a single linear map. Three standard choices are

$$\text{ReLU}(x) = \max(0, x), \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \text{SiLU}(x) = x \sigma(x), \quad (\text{A.5})$$

with $\sigma(x) = (1 + e^{-x})^{-1}$ the logistic sigmoid. The hyperbolic tangent is smooth and zero-centred but saturates for $|x| \gg 1$, where its gradient becomes vanishingly small and stalls training in deeper networks. ReLU [35] is computationally cheap and avoids this saturation, but has a hard zero gradient on the negative half-axis, which can leave units permanently inactive. The Sigmoid Linear Unit (SiLU) [17], also known as Swish, replaces this hard cut-off by a smooth, self-gated profile that retains a non-zero gradient for $x < 0$ while approaching ReLU for large positive x . Both the surrogate and the encoder/decoder networks of this thesis use SiLU throughout, which gave more consistent convergence than ReLU.

Loss functions

For real-valued regression the mean squared error

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2 \quad (\text{A.6})$$

is the default; it is differentiable everywhere and penalises large deviations quadratically, but is correspondingly sensitive to outliers. The Huber loss [21]

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2, & |r| \leq \delta, \\ \delta(|r| - \frac{1}{2}\delta), & |r| > \delta, \end{cases} \quad r = y - \hat{y}, \quad (\text{A.7})$$

combines the smoothness of MSE near zero with the linear, outlier-robust tail of the absolute error. The surrogate of Chapter 4 uses the Huber loss on its compressed targets to dampen the influence of poorly conditioned components; the CVAE of Chapter 5 uses MSE on standardised geometry parameters as the reconstruction term inside its training objective.

Optimiser

Modern deep-learning practice has settled on first-order methods (using only the gradient, not the Hessian) of which the dominant variants are *adaptive*: they adjust the step size per parameter based on the gradient history. The most common ones, including the non-adaptive baseline SGD, are summarised in Table A.1. Both networks in this thesis use Adam [22], which maintains two running averages of past gradients: one of the gradients themselves, which smooths the update direction by carrying forward a fraction of the previous step (classical momentum), and one of their squares, which is used to scale the step size of each parameter separately. The per-parameter scaling means parameters with consistently large gradients take smaller steps and vice versa, which makes Adam relatively insensitive to the choice of base learning rate.

TABLE A.1: First-order stochastic optimisers commonly used to train MLPs.

Optimiser	Mechanism	Typical usage
SGD	Plain (mini-batch) stochastic gradient descent, optionally with momentum.	Large-scale problems where simplicity and well-tuned schedules suffice.
RMSprop	Per-parameter step size scaled by a running root-mean-square of recent gradients.	Non-stationary objectives, recurrent networks.
Adam [22]	Combines momentum on the gradient with an RMSprop-style adaptive step size.	General-purpose deep-learning default.
AdamW [31]	Adam with weight decay decoupled from the gradient update.	Deep networks where explicit L_2 regularisation matters.

Learning rate, batch size, and number of epochs

The learning rate η scales every parameter update and is, in practice, the most influential hyperparameter. A value that is too large causes the loss to oscillate or diverge; one that is too small drags out training and can leave the model in a poor local optimum. The base value of η is selected by validation-loss sweeps and is typically combined with a schedule that varies η during training. The batch size controls how many samples contribute to each gradient estimate: smaller batches give noisier but more frequent updates and tend to act as a mild regulariser, while larger batches stabilise the gradient direction and exploit hardware parallelism better. The number of epochs sets the maximum training budget; in this thesis it is paired with early stopping on the validation loss, so that the actual run length adapts to convergence rather than being fixed in advance.

Training techniques

Beyond the static hyperparameters above, five techniques shape the trajectory of training itself.

Early stopping. The validation loss is tracked at the end of every epoch, the best parameters seen so far are kept in memory, and training is halted once no improvement has been observed for a fixed patience window. Both networks use early stopping on the validation loss.

Learning-rate scheduling. Two strategies are used. *Plateau-based reduction* multiplies η by a fixed factor whenever the validation loss has not improved for a patience window. *Cosine annealing* [30] smoothly decays η from its base value to a small floor over a predefined number of epochs, biasing the optimisation towards broad exploration early and fine local refinement near the end. The surrogate uses the plateau scheme; the CVAE uses cosine annealing.

Gradient clipping. Norm-based clipping rescales the gradient $\mathbf{g}$ whenever its norm exceeds a threshold c :

$$\mathbf{g} \leftarrow \mathbf{g} \cdot \min\left(1, \frac{c}{\|\mathbf{g}\|}\right), \quad (\text{A.8})$$

which preserves the gradient direction while bounding its magnitude [38]. This is applied to the CVAE, whose composite training objective occasionally produces large transient gradients.

Layer normalisation. Layer normalisation [2] subtracts the mean and divides by the standard deviation of each layer’s pre-activations within a single sample, then applies a learned affine transform. Unlike batch normalisation it does not depend on batch statistics and is therefore stable for the small mini-batches used at validation and inference. It is applied between the affine and nonlinear parts of every encoder and decoder layer of the CVAE.

Residual connections. A residual block [16] adds the input of a sub-network back to its output,

$$\mathbf{y} = f_{\text{block}}(\mathbf{x}) + \mathbf{x}, \quad (\text{A.9})$$

so the block only has to learn the deviation from identity. The skip path provides a direct route for the gradient during back-propagation, which makes deeper networks practical to train. The performance surrogate of Chapter 4 stacks several such blocks; the CVAE encoder and decoder use plain MLP layers without skips.

A.6 Variational Autoencoder: Full Derivation

Section 2.3.2 states the evidence lower bound used in this thesis without retracing its derivation. The standard from-first-principles route is reproduced here for completeness; the development follows Kingma and Welling [24].

The VAE models the data as the marginal of a latent-variable model

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z}, \quad (\text{A.10})$$

with prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a decoder $p_\theta(\mathbf{x} | \mathbf{z})$ parameterised by a neural network. Direct maximisation of $\log p_\theta(\mathbf{x})$ is intractable: the integral has no closed form, and Monte-Carlo estimates have high variance because most samples drawn from the prior do not explain a given observation.

Variational lower bound. A second network, the encoder $q_\phi(\mathbf{z} | \mathbf{x})$, is introduced as a tractable approximation to the true posterior $p_\theta(\mathbf{z} | \mathbf{x})$. Multiplying and dividing the integrand by q_ϕ inside the expectation yields the standard decomposition

$$\log p_\theta(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] + \text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| p_\theta(\mathbf{z} | \mathbf{x})), \quad (\text{A.11})$$

where the Kullback–Leibler divergence

$$\text{KL}(q \| p) = \mathbb{E}_q[\log q(\mathbf{z}) - \log p(\mathbf{z})] \geq 0 \quad (\text{A.12})$$

quantifies the dissimilarity between two distributions, expressed in *nats* when the natural logarithm is used (1 nat = $\log_2 e \approx 1.44$ bits). Because the KL term in (A.11) is non-negative, the first term is a lower bound on $\log p_\theta(\mathbf{x})$, called the *evidence lower bound* (ELBO):

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - \text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z})). \quad (\text{A.13})$$

Maximising (A.13) jointly over θ and ϕ pushes $\log p_\theta(\mathbf{x})$ upward while simultaneously driving q_ϕ towards the true posterior.

The two terms. Under a Gaussian decoder $p_\theta(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\hat{\mathbf{x}}_\theta(\mathbf{z}), \sigma_x^2 \mathbf{I})$, taking the log of the reconstruction term reduces (up to constants) to the negative squared error between $\mathbf{x}$ and the decoder output $\hat{\mathbf{x}}_\theta(\mathbf{z})$, recovering the MSE loss of §A.5. The KL term is a *regulariser* that pulls the encoder posterior towards the prior. For diagonal Gaussian posteriors $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$ and the standard prior, the KL admits the closed form

$$\text{KL}(q_\phi(\mathbf{z} | \mathbf{x}) \| \mathcal{N}(\mathbf{0}, \mathbf{I})) = \frac{1}{2} \sum_{j=1}^d (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1), \quad (\text{A.14})$$

which is differentiable in $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and is contributed to the loss directly. This closed form is the version used in the main text and in the CVAE loss of Chapter 5.

Appendix B

Dataset and Preprocessing

This appendix complements Chapter 3. Section B.1 lists five representative ($\mathbf{G}, \mathbf{C}$) pairs drawn from the working dataset to make the data layout concrete. Section B.2 collects the two supplementary EDA figures referenced from §3.3.1.

B.1 Representative Dataset Rows

Tables B.1 and B.2 show five rows drawn at evenly spaced indices (rows 1000, 20 000, 40 000, 60 000, and 80 000) from the post-filter working dataset of 85 384 samples. Each row index is shared between the two tables, so row 1 of Table B.1 and row 1 of Table B.2 describe the same motor design. Values are in raw (un-normalised) units.

TABLE B.1: Five representative geometry vectors $\mathbf{G}$ from the working dataset, split across two row-blocks for readability.

#	DSO	GAP	KAR	KBETA	KFD	KGAMMA	KMQ1	KMQ2
1	0.191	0.807	0.844	0.343	1.382	0.133	0.775	0.467
2	0.212	1.001	0.769	0.213	2.531	0.257	0.546	0.415
3	0.237	0.848	0.721	0.252	1.409	0.282	0.581	0.843
4	0.202	0.650	1.053	0.223	1.576	0.055	0.559	0.445
5	0.232	0.502	0.752	0.322	1.876	0.172	0.487	0.656

#	KYH	KPQ1	KPQ2	KQI	KQIVAR	KR0	KR1	KSR	KTW
1	0.469	0.605	0.432	0.579	0.417	0.606	0.507	0.733	0.561
2	0.416	0.611	0.343	0.512	0.529	0.655	0.358	0.652	0.448
3	0.492	0.855	0.834	0.740	0.417	0.591	0.461	0.712	0.545
4	0.494	0.136	0.174	0.652	0.510	0.555	0.655	0.700	0.527
5	0.476	0.675	0.852	0.546	0.385	0.618	0.570	0.777	0.533

B. DATASET AND PREPROCESSING

TABLE B.2: Condition vectors $\mathbf{C}$ corresponding to the five geometry rows in Table B.1.

#	ω_{OP1} (rpm)	T_{OP1} (Nm)	ω_{OP3} (rpm)	T_{OP3} (Nm)	I_{max} (A)	U_{max} (V)	N
1	9 250	75.6	14 000	16.0	219.8	117.6	1
2	1 250	390.6	20 000	14.9	653.7	107.1	4
3	14 750	118.0	20 000	90.3	302.9	222.0	1
4	2 250	349.2	20 000	42.6	275.9	134.8	3
5	5 250	348.7	20 000	82.7	409.0	192.9	2

B.2 Supplementary EDA Figures

This section collects the two supporting figures referenced from the exploratory analysis of §3.3.1. Figure B.1 reports the full 7×17 Spearman correlation matrix between the conditions and the geometry parameters; the three columns whose strongest correlation with any component of $\mathbf{C}$ is largest in absolute value (KAR, DSO, KMQ1) are the ones retained in the pair-matrix of Figure 3.5. Figure B.2 reports the corresponding 17×17 pair-matrix between all geometry parameters, with the same plotting conventions as Figure 3.5.

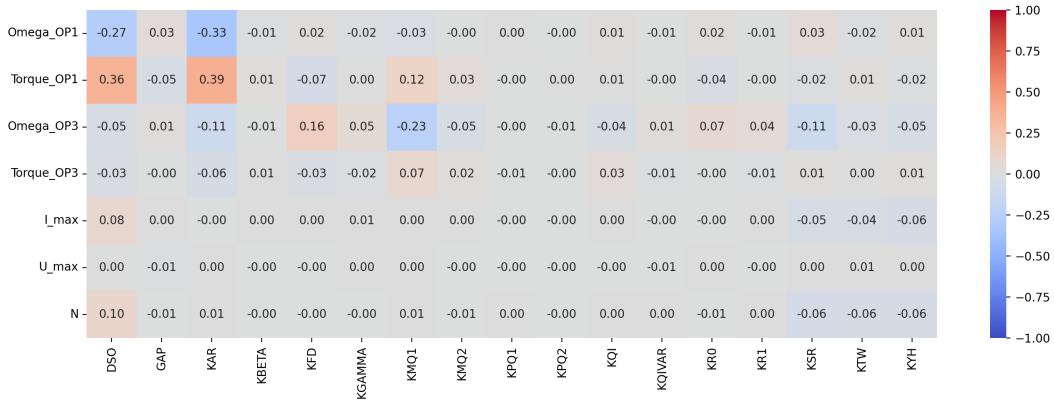


FIGURE B.1: Spearman correlation between each condition (rows) and each geometry parameter (columns) on the post-filter dataset ($M = 85\,384$). The three columns most strongly correlated with $\mathbf{C}$ (KAR, DSO, KMQ1) are selected for the pair-matrix in Figure 3.5.

B.2. Supplementary EDA Figures

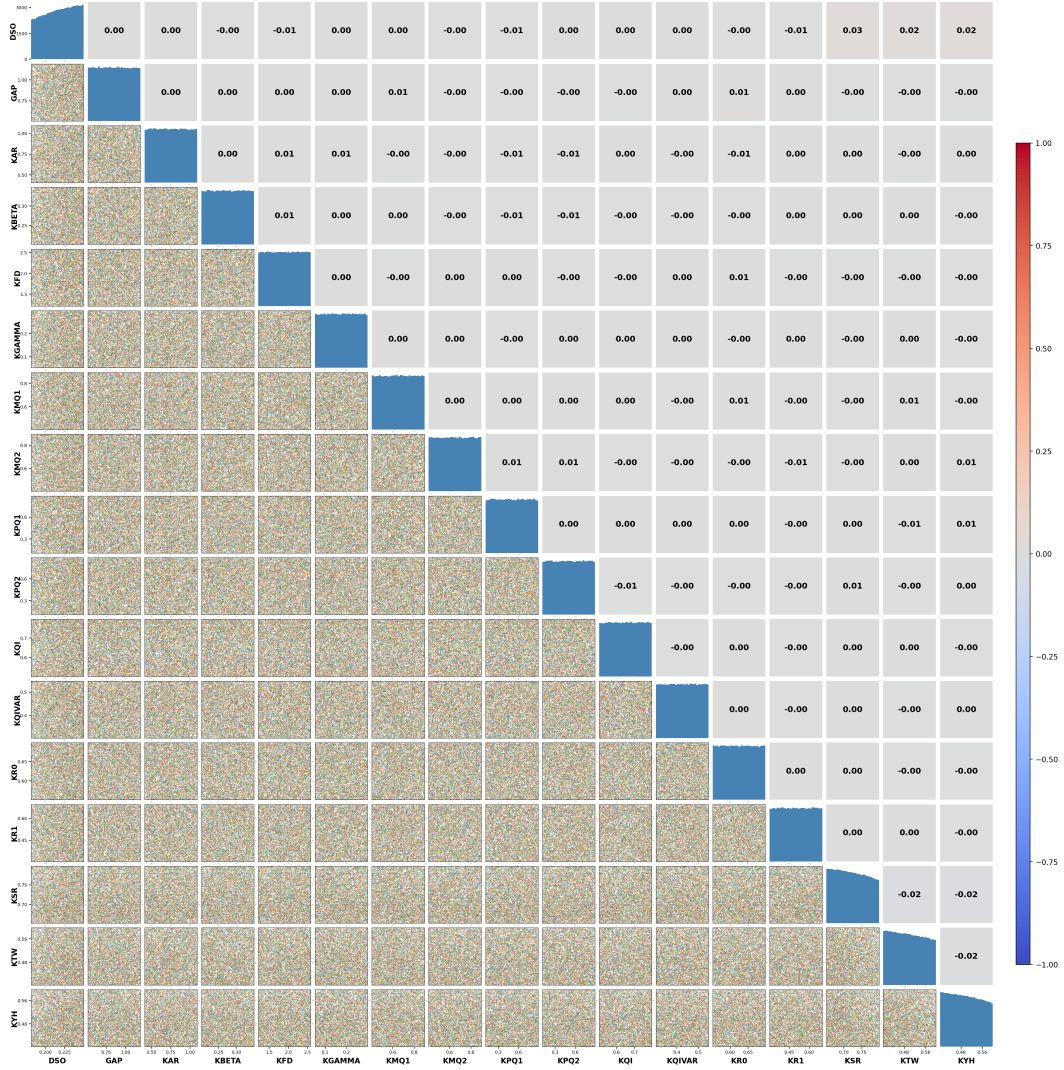


FIGURE B.2: Pair-matrix between all 17 geometry parameters; cell conventions are identical to Figure 3.5. The absence of any significant off-diagonal correlation ($|\rho_S| < 0.05$ throughout) confirms that the LHS marginals remain effectively independent after feasibility filtering.

Appendix C

CVAE Evaluation Details

This appendix collects the supporting figures and tables that complement the CVAE evaluation of Chapter 5, §5.5.

C.1 Per-feature reconstruction metrics

Table C.1 reports MAE, RMSE, R^2 and MAPE per geometric parameter for both CVAE variants on the test set; errors are computed in physical units after inverting the min-max normalisation of §5.1.3. Figure C.1 shows the corresponding 17-panel parity grid.

TABLE C.1: Per-feature reconstruction error on the test set, for both CVAE variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent.

Feature	Standard				Regularised			
	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	MAPE
DSO	0.00459	0.00574	0.918	2.12	0.00471	0.00603	0.909	2.18
GAP	0.05192	0.06490	0.896	6.77	0.04529	0.05688	0.920	5.89
KAR	0.03346	0.04301	0.955	4.73	0.03459	0.04461	0.951	4.87
KBETA	0.00873	0.01177	0.925	3.30	0.00849	0.01122	0.932	3.19
KFD	0.08642	0.10936	0.926	5.00	0.09442	0.12300	0.907	5.45
KGAMMA	0.01540	0.01999	0.923	12.57	0.01535	0.01986	0.924	12.47
KMQ1	0.02919	0.03873	0.928	4.60	0.03771	0.05123	0.875	6.05
KMQ2	0.02976	0.03886	0.928	4.93	0.02972	0.03923	0.926	4.97
KPQ1	0.04666	0.06280	0.926	16.17	0.04635	0.06278	0.926	16.26
KPQ2	0.04620	0.06151	0.929	16.34	0.04760	0.06210	0.928	14.65
KQI	0.01866	0.02402	0.923	2.96	0.01836	0.02387	0.924	2.90
KQIVAR	0.01395	0.01906	0.929	3.46	0.01432	0.01863	0.933	3.57
KR0	0.00902	0.01176	0.927	1.46	0.00915	0.01195	0.925	1.47
KR1	0.02340	0.03101	0.928	5.07	0.02343	0.03161	0.925	5.20
KSR	0.01070	0.01327	0.907	1.48	0.01009	0.01286	0.912	1.40
KTW	0.01306	0.01663	0.916	2.65	0.01294	0.01632	0.919	2.64
KYH	0.01323	0.01692	0.913	2.73	0.01268	0.01632	0.919	2.59

C. CVAE EVALUATION DETAILS

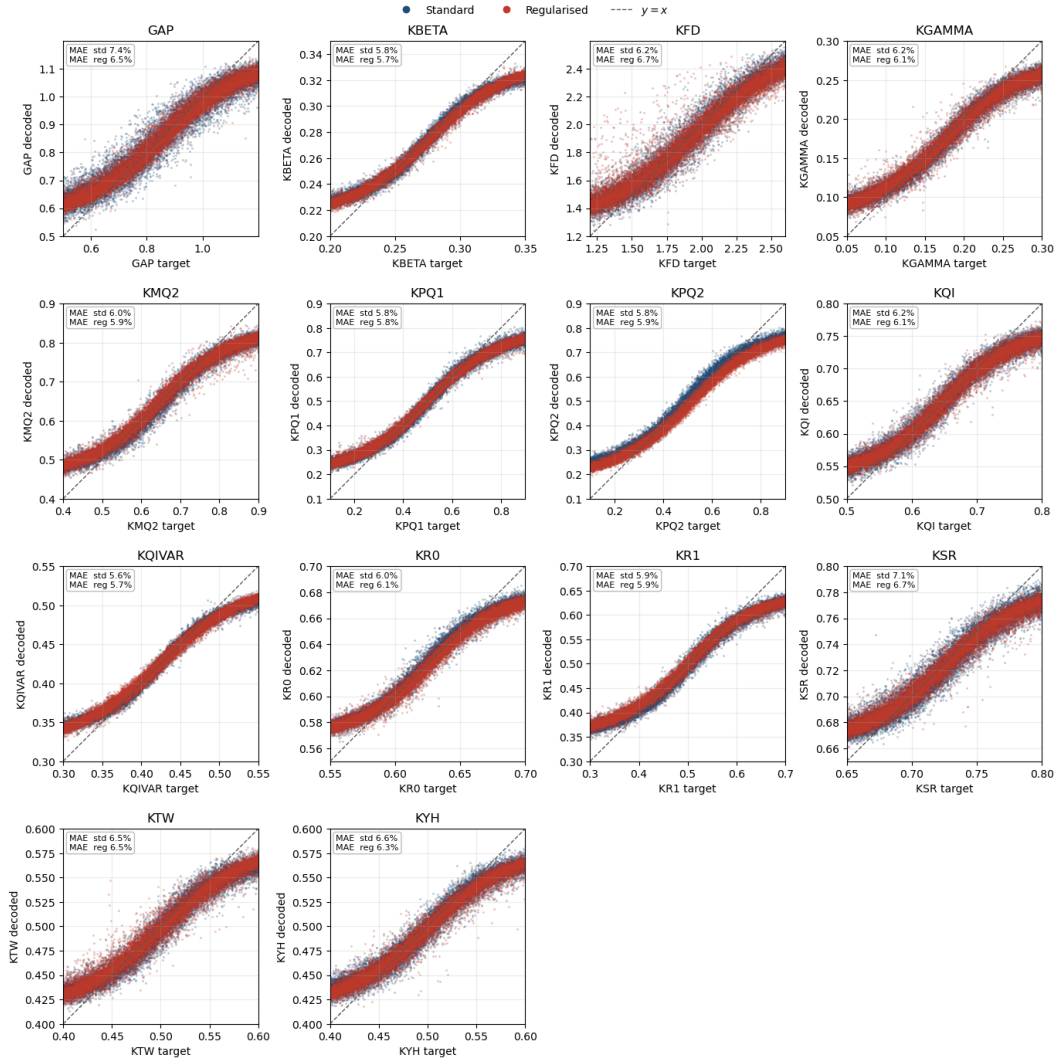


FIGURE C.1: Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and regularised CVAE.

One feature-level observation complements Table C.1 beyond the discussion in §5.5.1: KAR attains the highest R^2 (≈ 0.95 for both variants), in line with its role as one of the most condition-sensitive parameters identified in §3.3.1 (the decoder picks up the conditioning signal most strongly along this axis). The inflated MAPE values of KGAMMA, KPQ1 and KPQ2 (12–16 %) and the localised reconstruction loss on KMQ1 under the performance regulariser (MAE $0.029 \rightarrow 0.038$) are discussed in the main text.

C.2 Latent-space diagnostics

Figures C.2 and C.3 show the empirical marginal posterior $q(z_k)$ on the test set for each of the 15 latent dimensions, overlaid with the prior $\mathcal{N}(0, 1)$, for the standard and regularised CVAE respectively.

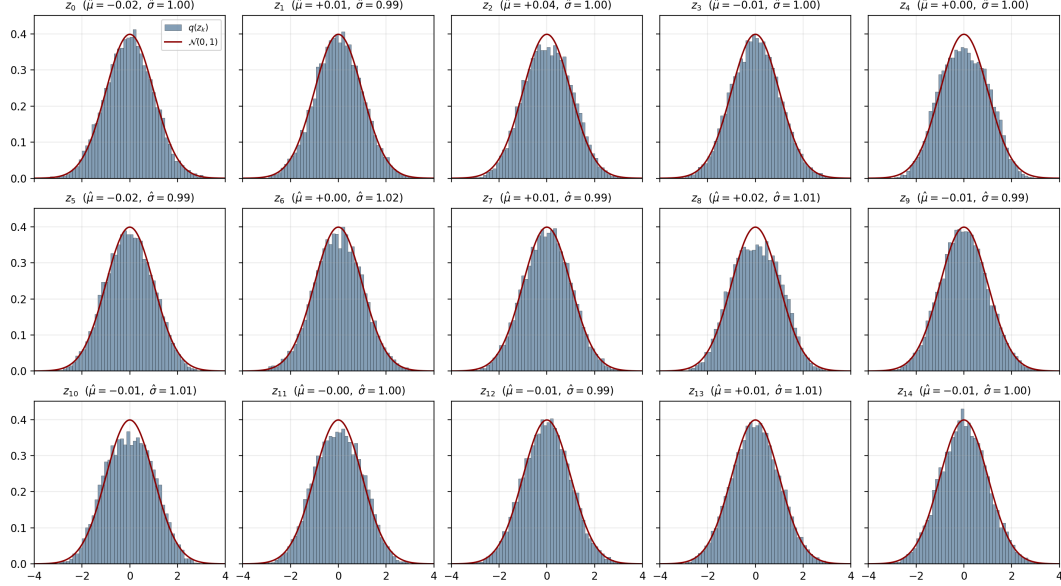


FIGURE C.2: Marginal posteriors of the 15 latent dimensions of the standard CVAE on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

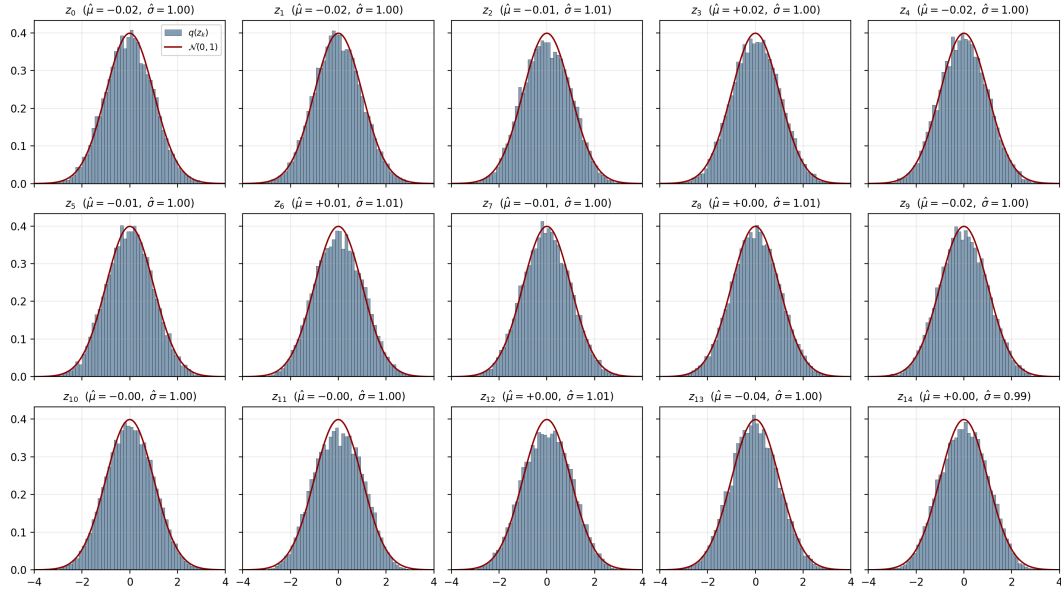


FIGURE C.3: Marginal posteriors of the 15 latent dimensions of the performance-regularised CVAE on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

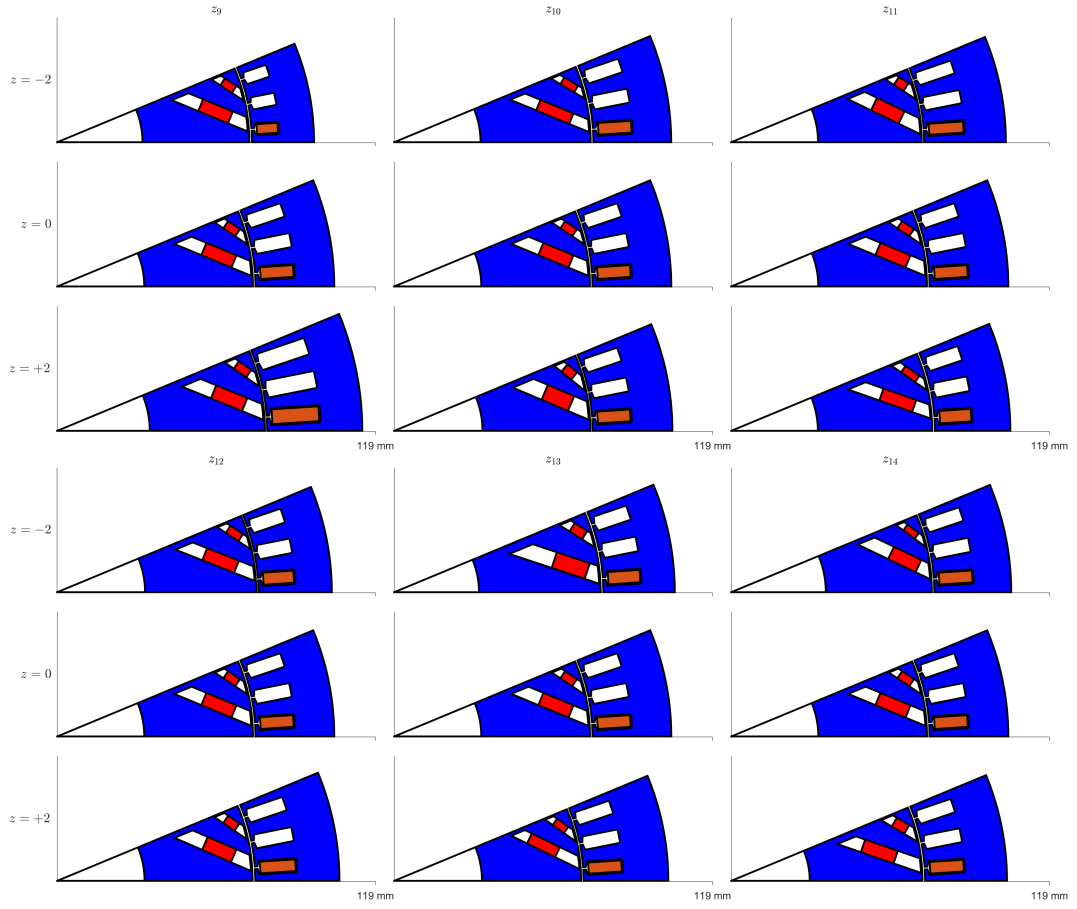


FIGURE C.4: (continued).

C. CVAE EVALUATION DETAILS

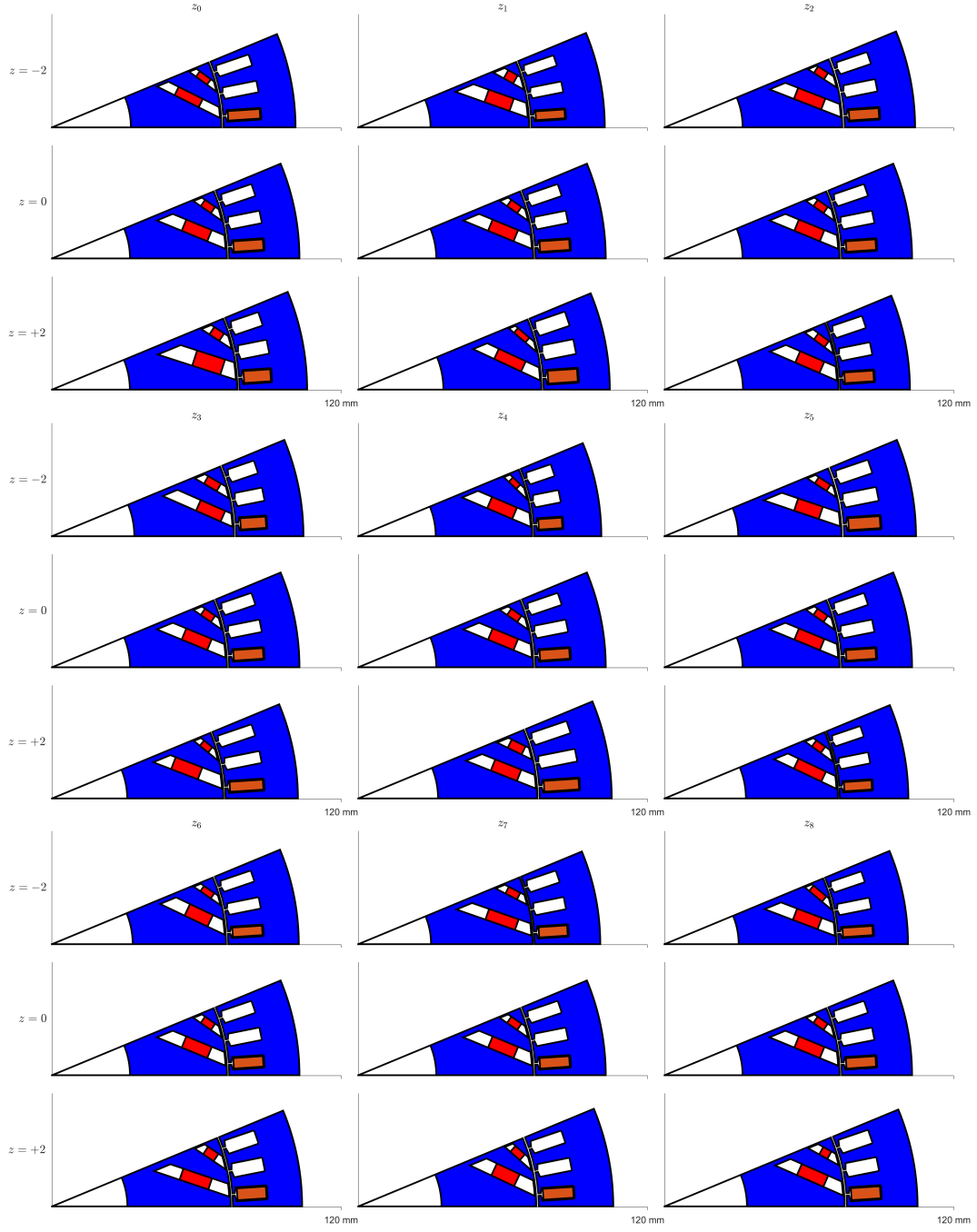


FIGURE C.5: Latent traversals along all 15 dimensions of the performance-regularised CVAE.

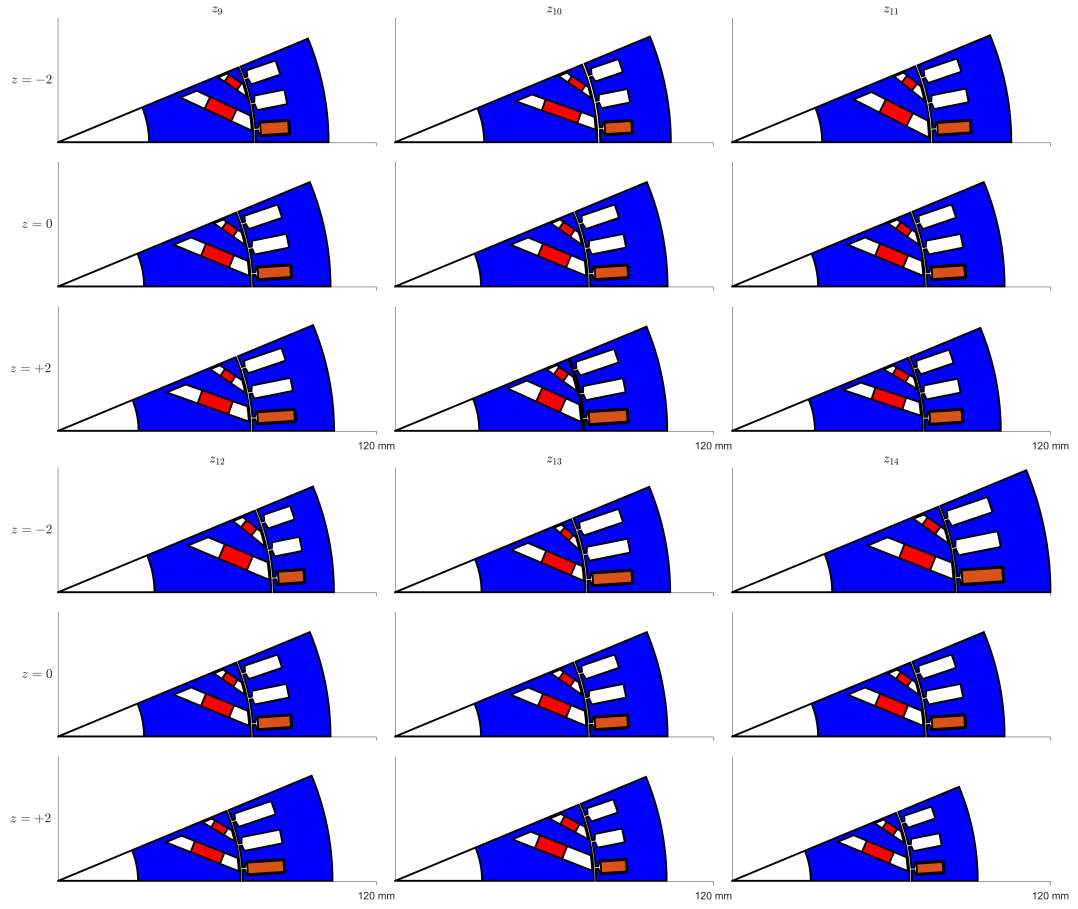


FIGURE C.5: (continued).

C.3 Alternative reconstruction training objectives

The parity plots of Figures 5.4 and C.1 show that residual magnitudes increase systematically as the normalised target x_j approaches the lower or upper edge of $[0, 1]$. This subsection links this effect to the interaction between the sigmoid output of the decoder and the mean-squared reconstruction term, and then introduces two established alternative training objectives that mitigate this attenuation effect. The two alternatives are evaluated on the same model-internal diagnostics as the baseline so that they can be compared to the results of §C.1–§C.2. A full downstream comparison through the case study and the latent optimisation falls outside the scope of this work and is left as future work. The cost function and the output unit are closely linked design choices: in most cases the cost is the cross-entropy between the data distribution and the model distribution, and the form of that cross-entropy follows from how the output is represented [14].

Let $\hat{\ell}_j$ denote the per-element pre-sigmoid logit produced by the decoder, so that $\hat{x}_j = \sigma(\hat{\ell}_j)$ is the bounded output of §5.1 and $\epsilon_j = \hat{x}_j - x_j$ is the per-element residual against the normalised target $x_j \in [0, 1]$. Per element, the mean-squared reconstruction term reads

$$L_{\text{MSE}}(\hat{\ell}_j, x_j) = (\sigma(\hat{\ell}_j) - x_j)^2, \quad (\text{C.1})$$

and its gradient with respect to the logit follows from the chain rule:

$$\frac{\partial L_{\text{MSE}}}{\partial \hat{\ell}_j} = 2\epsilon_j \cdot \underbrace{\sigma(\hat{\ell}_j)(1 - \sigma(\hat{\ell}_j))}_{\sigma'(\hat{\ell}_j)}. \quad (\text{C.2})$$

The gradient is proportional to the residual ϵ_j , but multiplied by an attenuation factor $\sigma'(\hat{\ell}_j) = \sigma(\hat{\ell}_j)(1 - \sigma(\hat{\ell}_j))$, which tends to zero as $\hat{x}_j \rightarrow 0$ or $\hat{x}_j \rightarrow 1$. Targets near the domain extremes therefore push the decoder with a smaller learning signal than targets in the bulk of $[0, 1]$, even when the residual itself is large.

The two subsections below explore two alternative training objectives that mitigate this attenuation. In both cases, no modifications were made to the model apart from the training objective itself (i.e., the same hyperparameters were used), in order to keep the comparison tractable.

C.3.1 BCE with logits

Binary cross-entropy with logits minimises the cross-entropy between $p = \text{Bernoulli}(x_j)$ and $q = \text{Bernoulli}(\sigma(\hat{\ell}_j))$, evaluated directly from the logit in a numerically stable form:

$$L_{\text{BCE}}(\hat{\ell}_j, x_j) = -\left[x_j \log \sigma(\hat{\ell}_j) + (1 - x_j) \log(1 - \sigma(\hat{\ell}_j)) \right]. \quad (\text{C.3})$$

The logit gradient simplifies to

$$\frac{\partial L_{\text{BCE}}}{\partial \hat{\ell}_j} = \sigma(\hat{\ell}_j) - x_j = \epsilon_j, \quad (\text{C.4})$$

i.e. the σ' attenuation of (C.2) disappears and the gradient at the logit equals the residual itself. In practice the loss is implemented through the softplus identity $\log \sigma(\hat{\ell}_j) = -\zeta(-\hat{\ell}_j)$ with $\zeta(u) = \log(1+e^u)$, so that $\sigma(\hat{\ell}_j)$ never needs to be evaluated explicitly. Were it to underflow to zero, its logarithm would diverge to $-\infty$ and corrupt the loss [14, 40]. Only the training loss changes: the decoder still terminates in a sigmoid, so the sampling pipeline of §5.4 is unchanged. The diagnostics below mirror those of §C.1–§C.2, here applied to the standard and performance-regularised BCE-with-logits variants.

Reconstruction Quality

TABLE C.2: Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for the standard and performance-regularised BCE-with-logits variants.

Variant	MAE [% of range]	RMSE [% of range]	R^2	MAPE [%]
Standard BCE-with-logits	2.45	3.10	0.988	2.18
Regularised BCE-with-logits	2.30	3.12	0.987	1.97

TABLE C.3: Per-feature reconstruction error on the test set for the standard and performance-regularised BCE-with-logits variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent.

Feature	Standard BCE-with-logits				Regularised BCE-with-logits			
	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	MAPE
DSO	0.00206	0.00264	0.983	0.95	0.00176	0.00241	0.985	0.82
GAP	0.02029	0.02483	0.985	2.57	0.01644	0.02101	0.989	2.06
KAR	0.01557	0.02101	0.989	2.21	0.01610	0.02195	0.988	2.22
KBETA	0.00330	0.00410	0.991	1.25	0.00281	0.00375	0.992	1.06
KFD	0.03899	0.05068	0.984	2.23	0.04417	0.06332	0.975	2.52
KGAMMA	0.00592	0.00736	0.990	4.83	0.00551	0.00736	0.990	4.39
KMQ1	0.01429	0.02036	0.980	2.28	0.02092	0.02990	0.957	3.36
KMQ2	0.01126	0.01391	0.991	1.87	0.01023	0.01377	0.991	1.66
KPQ1	0.01749	0.02196	0.991	5.64	0.01372	0.01831	0.994	4.37
KPQ2	0.01753	0.02186	0.991	5.68	0.01407	0.01852	0.994	4.34
KQI	0.00734	0.00907	0.989	1.16	0.00656	0.00848	0.990	1.03
KQIVAR	0.00548	0.00679	0.991	1.35	0.00447	0.00587	0.993	1.10
KR0	0.00349	0.00432	0.990	0.56	0.00329	0.00435	0.990	0.53
KR1	0.00879	0.01094	0.991	1.92	0.00770	0.01021	0.992	1.67
KSR	0.00416	0.00510	0.986	0.58	0.00360	0.00482	0.988	0.50
KTW	0.00491	0.00604	0.989	1.01	0.00473	0.00644	0.987	0.97
KYH	0.00488	0.00600	0.989	1.00	0.00442	0.00601	0.989	0.90

C. CVAE EVALUATION DETAILS

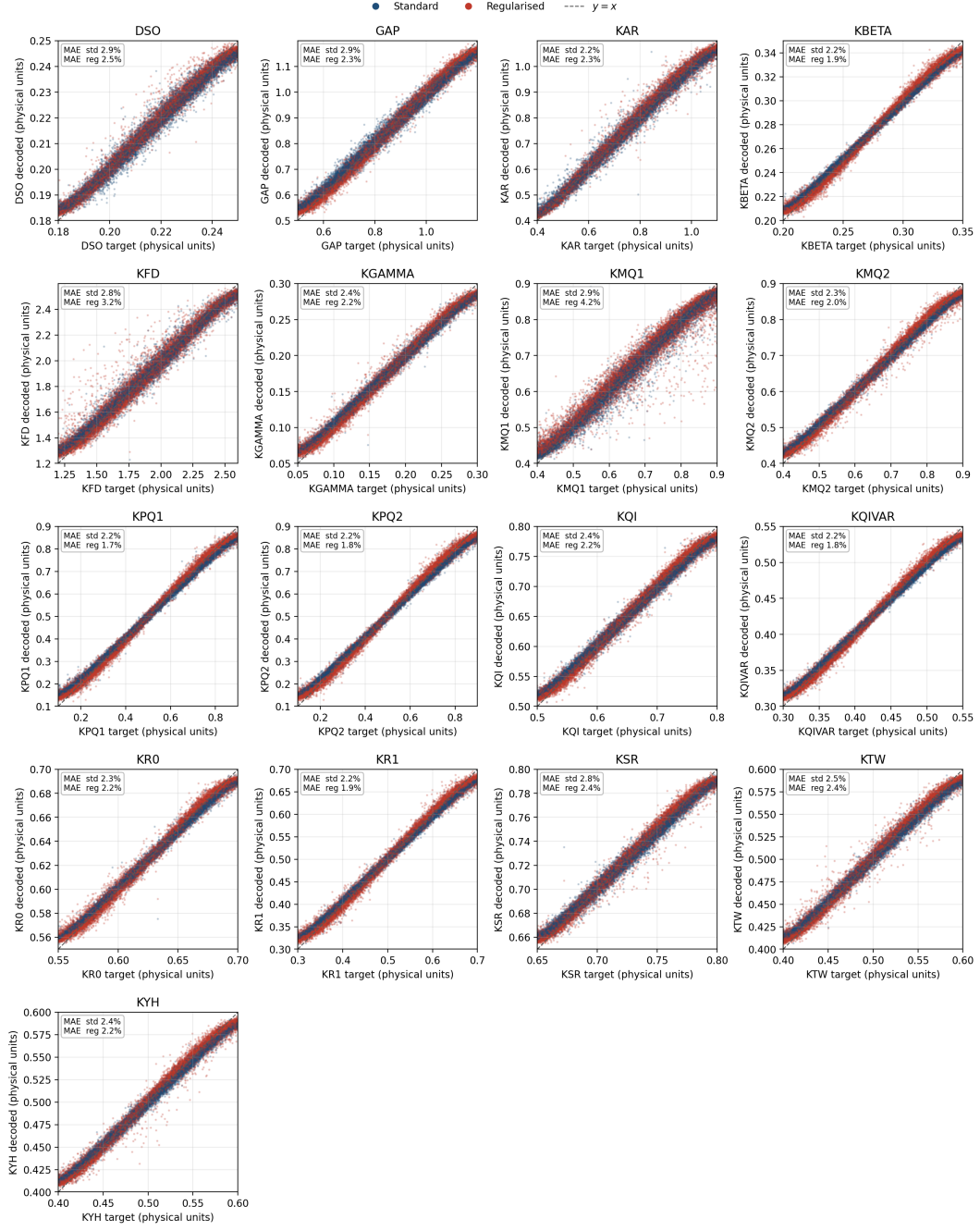


FIGURE C.6: Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and performance-regularised BCE-with-logits variants.

Latent Space Diagnostics

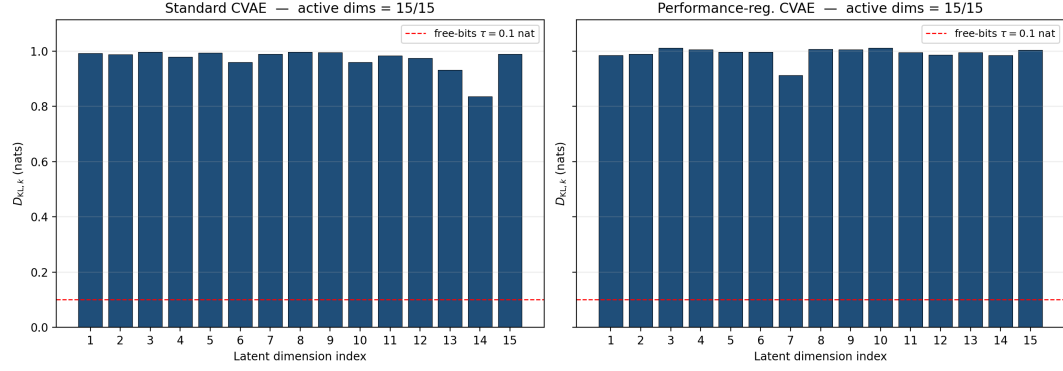


FIGURE C.7: Per-dimension KL divergence on the test set for the standard (left) and regularised (right) BCE-with-logits variant.

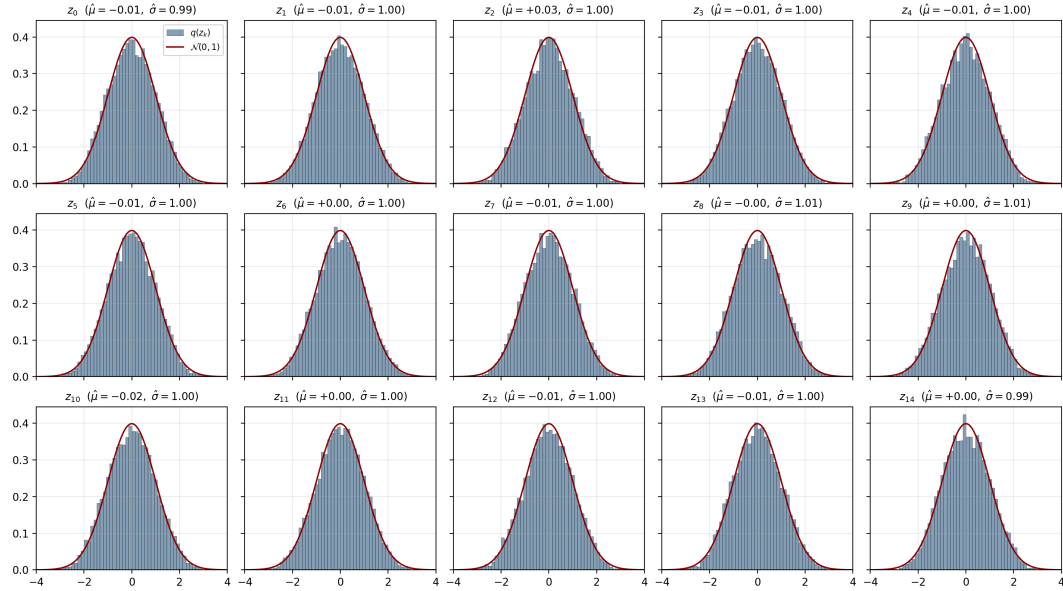


FIGURE C.8: Marginal posteriors of the 15 latent dimensions of the standard BCE-with-logits variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

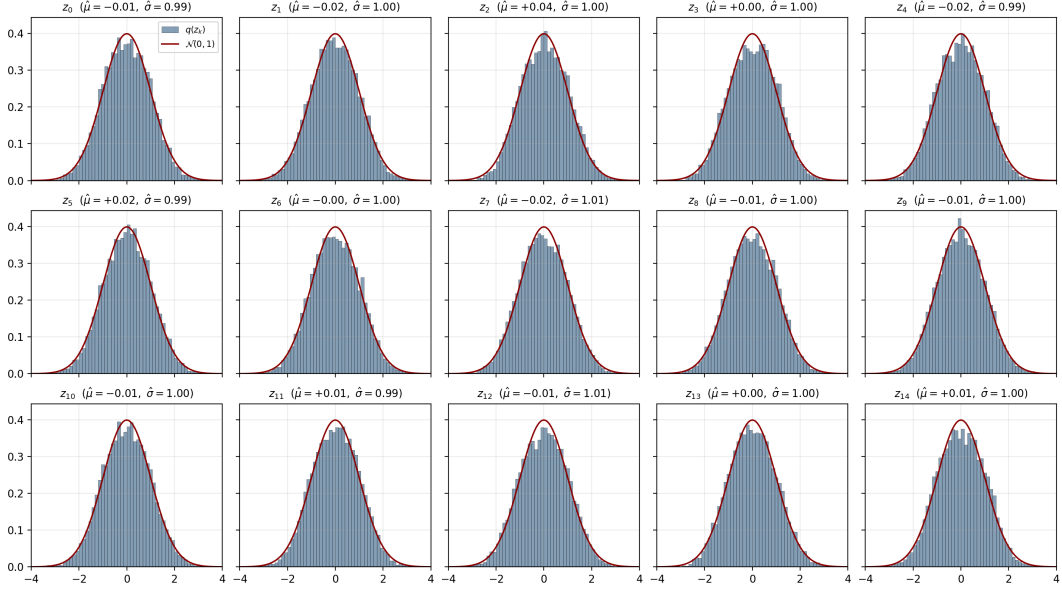


FIGURE C.9: Marginal posteriors of the 15 latent dimensions of the performance-regularised BCE-with-logits variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

C.3.2 Continuous Bernoulli

The continuous Bernoulli distribution $\mathcal{CB}(\lambda)$ is the exponential-family extension of the Bernoulli to the interval $[0, 1]$, introduced by Loaiza-Ganem and Cunningham [29] precisely to remove the inconsistency that arises when BCE is used as a log-likelihood on continuous targets. The decoder predicts a parameter $\lambda_j = \sigma(\hat{\ell}_j) \in (0, 1)$ and the negative log-likelihood per element reads

$$L_{\text{CB}}(\hat{\ell}_j, x_j) = -\log C(\lambda_j) - x_j \log \lambda_j - (1 - x_j) \log(1 - \lambda_j), \quad (\text{C.5})$$

with normalising constant

$$C(\lambda) = \begin{cases} \frac{2 \operatorname{arctanh}(1 - 2\lambda)}{1 - 2\lambda} & \lambda \neq \frac{1}{2}, \\ 2 & \lambda = \frac{1}{2}. \end{cases} \quad (\text{C.6})$$

As with BCE-with-logits, the σ' attenuation at the logit disappears; the extra $-\log C(\lambda_j)$ term corrects the bias that would otherwise arise from using BCE as a log-likelihood on continuous $x_j \in [0, 1]$. The standard sigmoid-cross-entropy motivation of [14] concerns genuinely binary targets; on continuous $[0, 1]$ -targets BCE alone is no longer a proper log-likelihood, which is precisely the gap that the $-\log C(\lambda_j)$ correction closes. Both the training objective and the output decoding change: at inference the decoder no longer returns $\sigma(\hat{\ell})$ but the CB mean $\mu(\lambda)$, which is taken as the reconstructed normalised geometry before inverting the min-max scaling of §5.1.3. The diagnostics below again mirror those of §C.1–§C.2, here applied to the standard and performance-regularised continuous-Bernoulli variants.

Reconstruction QualityTABLE C.4: Reconstruction error on the test set, aggregated over the 17 geometric parameters of $\mathbf{G}$, for the standard and performance-regularised continuous-Bernoulli variants.

Variant	MAE [% of range]	RMSE [% of range]	R^2	MAPE [%]
Standard continuous Bernoulli	1.43	2.01	0.995	1.28
Regularised continuous Bernoulli	2.24	2.97	0.989	1.89

TABLE C.5: Per-feature reconstruction error on the test set for the standard and performance-regularised continuous-Bernoulli variants. MAE and RMSE are in physical units; R^2 is dimensionless; MAPE is in percent.

Feature	Standard continuous Bernoulli				Regularised continuous Bernoulli			
	MAE	RMSE	R^2	MAPE	MAE	RMSE	R^2	MAPE
DSO	0.00102	0.00141	0.995	0.47	0.00170	0.00232	0.987	0.79
GAP	0.00894	0.01226	0.996	1.16	0.01241	0.01621	0.994	1.57
KAR	0.01147	0.01643	0.993	1.68	0.01785	0.02331	0.987	2.57
KBETA	0.00180	0.00249	0.997	0.68	0.00265	0.00341	0.994	0.99
KFD	0.02720	0.03977	0.990	1.51	0.04409	0.06392	0.975	2.47
KGAMMA	0.00344	0.00475	0.996	2.71	0.00577	0.00732	0.990	4.10
KMQ1	0.01240	0.01858	0.984	2.00	0.01907	0.02672	0.966	3.08
KMQ2	0.00661	0.00915	0.996	1.10	0.00922	0.01213	0.993	1.53
KPQ1	0.00981	0.01359	0.997	3.29	0.01571	0.01972	0.993	4.53
KPQ2	0.00943	0.01325	0.997	3.02	0.01322	0.01728	0.994	4.07
KQI	0.00398	0.00545	0.996	0.63	0.00577	0.00759	0.992	0.91
KQIVAR	0.00300	0.00416	0.997	0.74	0.00443	0.00566	0.994	1.10
KR0	0.00211	0.00288	0.996	0.34	0.00297	0.00388	0.992	0.48
KR1	0.00495	0.00691	0.996	1.07	0.00727	0.00951	0.993	1.57
KSR	0.00216	0.00292	0.995	0.30	0.00462	0.00604	0.981	0.64
KTW	0.00271	0.00377	0.996	0.55	0.00423	0.00550	0.991	0.86
KYH	0.00255	0.00350	0.996	0.52	0.00413	0.00551	0.991	0.83

C. CVAE EVALUATION DETAILS

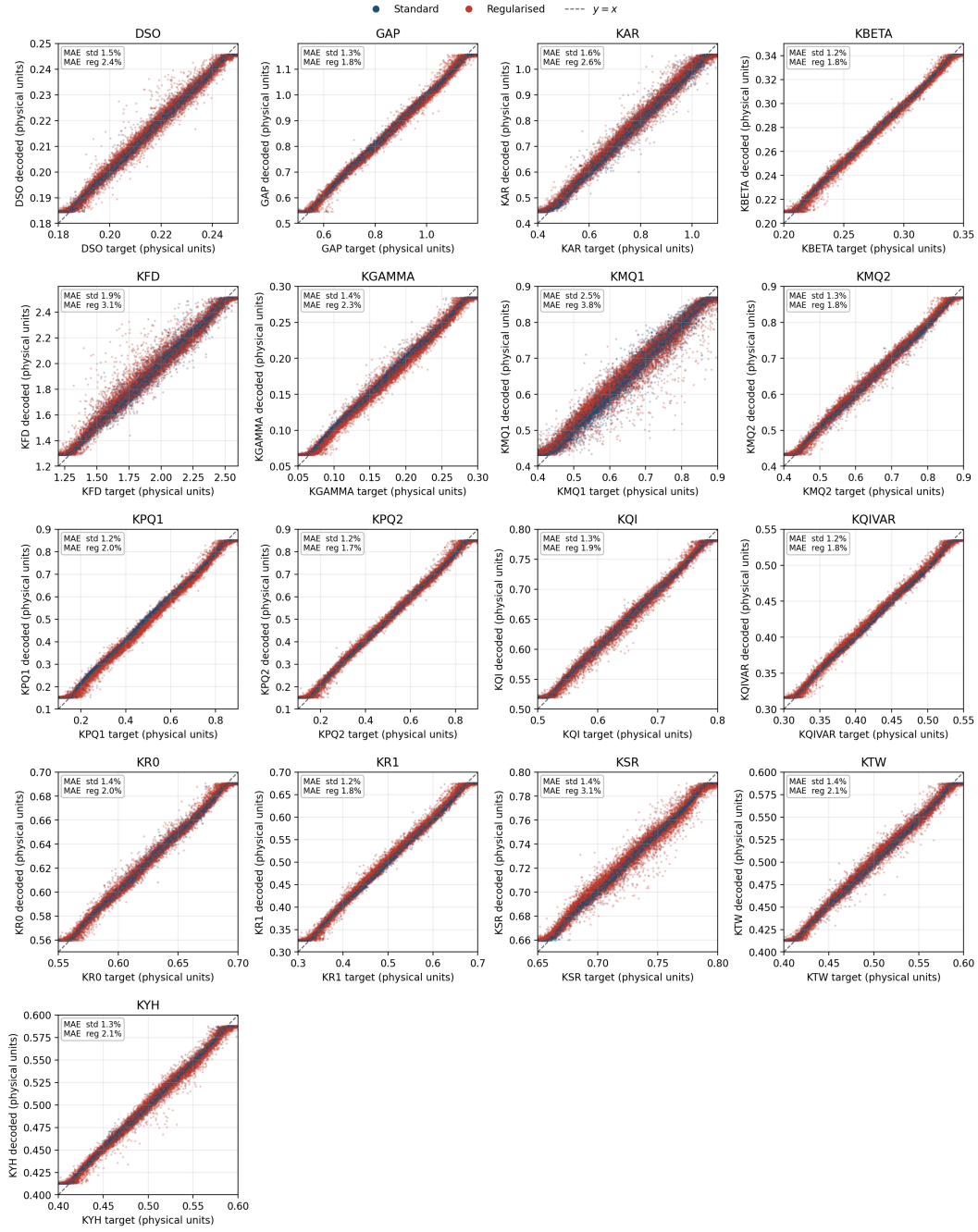


FIGURE C.10: Full 17-panel parity plot of decoded vs. target geometry features on the test set, for the standard and performance-regularised Continuous Bernoulli variants.

Latent Space Diagnostics

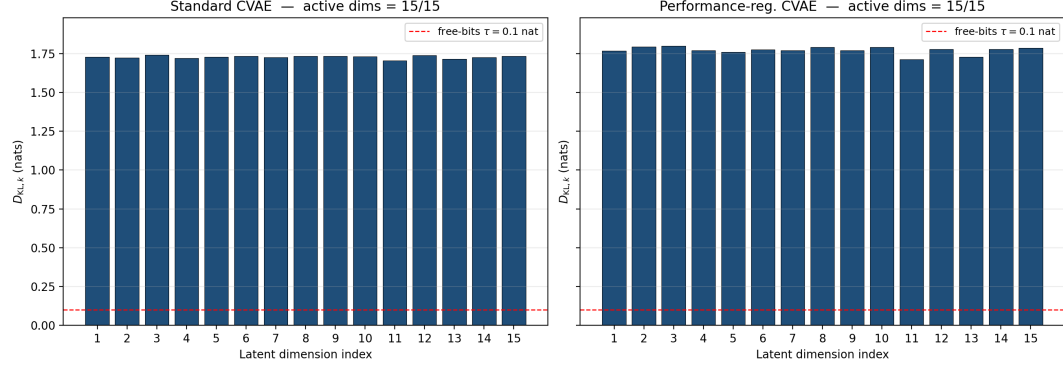


FIGURE C.11: Per-dimension KL divergence on the test set for the standard (left) and regularised (right) continuous-Bernoulli variant.

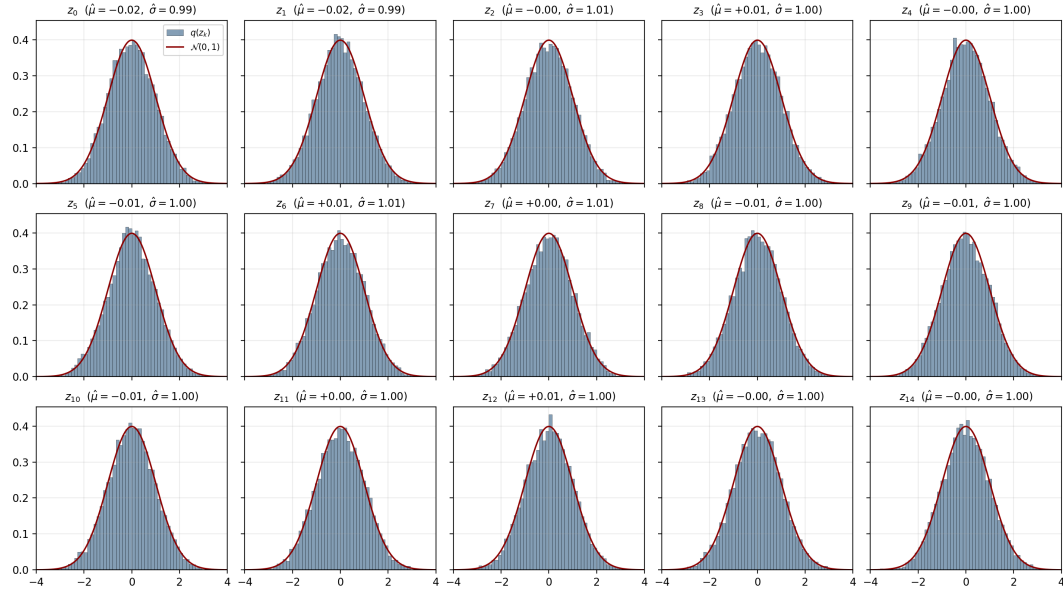


FIGURE C.12: Marginal posteriors of the 15 latent dimensions of the standard continuous-Bernoulli variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

C. CVAE EVALUATION DETAILS

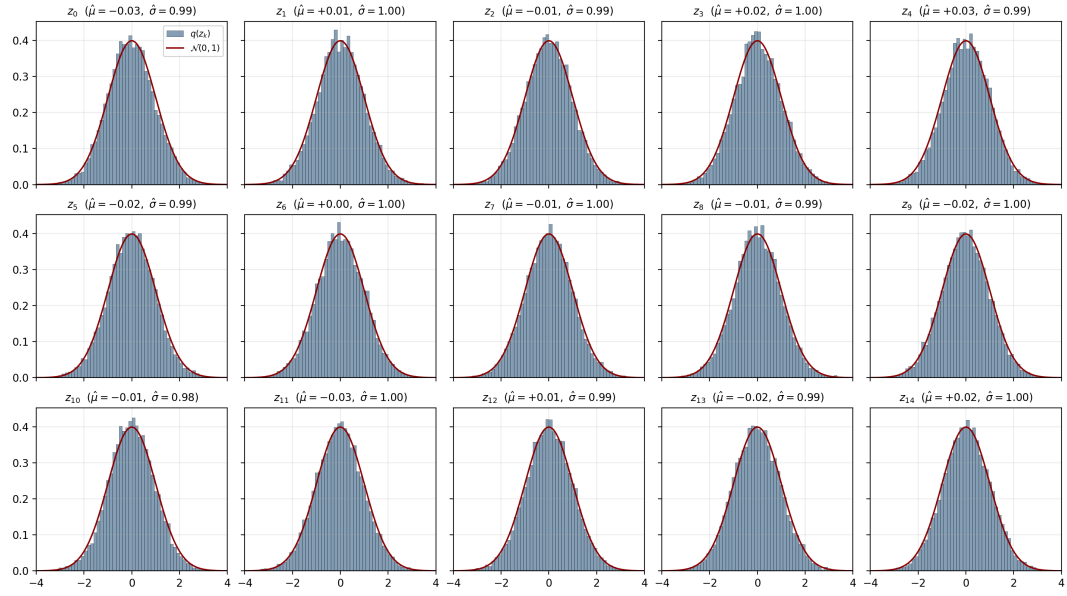


FIGURE C.13: Marginal posteriors of the 15 latent dimensions of the performance-regularised continuous-Bernoulli variant on the test set, with the prior $\mathcal{N}(0, 1)$ overlaid.

C.4 Case-study details

This section collects the supporting material for the case study of §5.5.3. Figure C.14 extends Figure 5.7 with the torque–speed curves of the remaining 4 industrial conditions. Figure C.15 extends Figure 5.8 with the marginal-distribution panels of the remaining 6 conditions. Table C.6 reports the per-feature range fraction for each of the 8 conditions and both variants; the last two rows reproduce the aggregate value of Table 5.3.

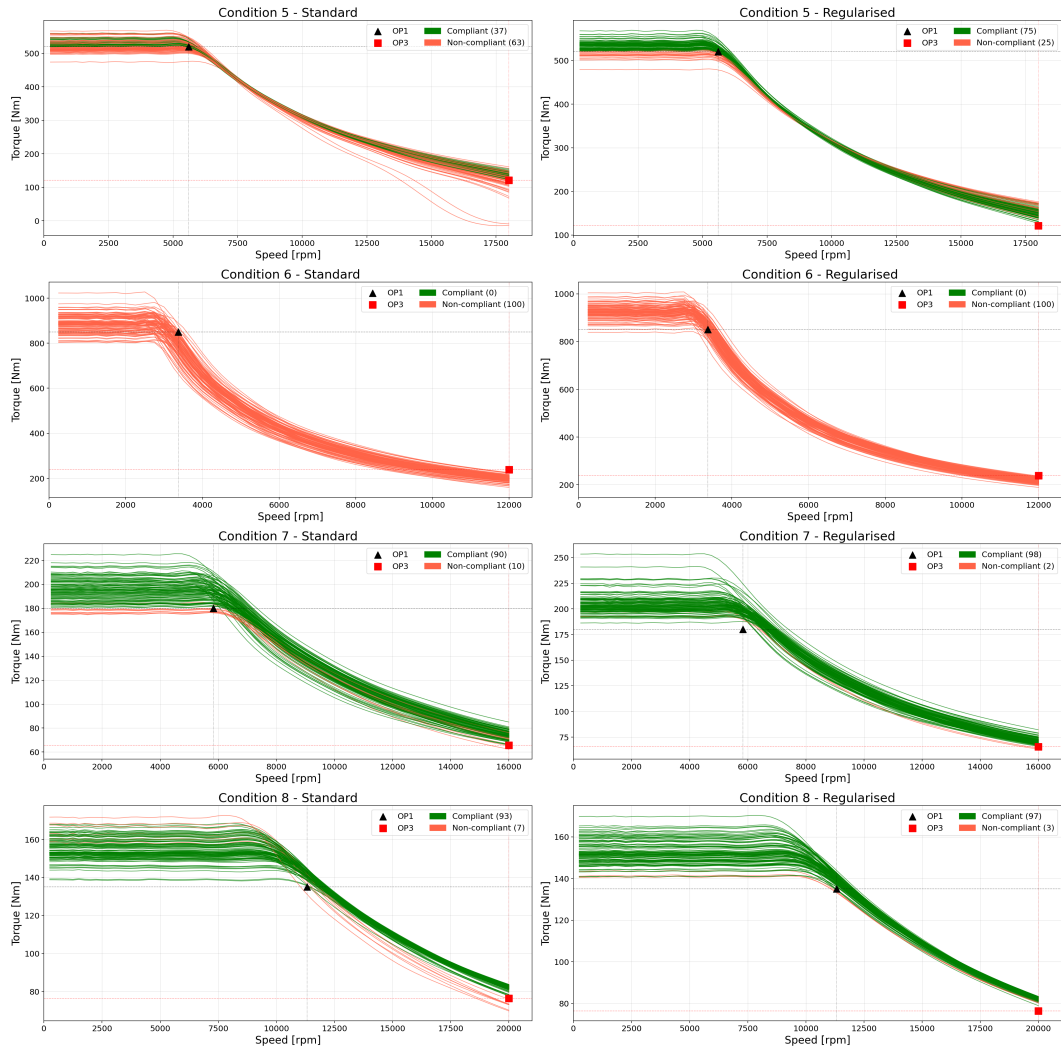


FIGURE C.14: Torque–speed curves of 100 candidate geometries generated for the remaining 4 industrial conditions of §5.5.3, evaluated through the MLP surrogate. Compliant samples are coloured green, non-compliant samples red; target operating points OP1 and OP3 are marked.

C. CVAE EVALUATION DETAILS

TABLE C.6: Per-feature range fraction [%] of the 1000 generated candidates per industrial condition (Conditions 1–8 of Table 5.2), for both CVAE variants (Std = standard, Reg = regularised). Averages over the 8 conditions are reported in the last two rows. Parameters DSO–KPQ1.

ID	DSO	GAP	KAR	KBETA	KFD	KGAMMA	KMQ1	KMQ2	KPQ1
1 Std	68	95	53	98	93	95	92	96	95
1 Reg	59	95	52	95	90	94	90	94	93
2 Std	56	95	34	95	93	95	93	95	93
2 Reg	43	96	31	96	91	92	85	94	94
3 Std	73	94	54	96	95	96	93	94	97
3 Reg	72	94	54	95	90	94	87	95	93
4 Std	63	95	19	95	88	92	85	95	95
4 Reg	72	96	20	95	90	91	81	95	92
5 Std	63	89	61	94	66	91	61	91	93
5 Reg	75	95	53	97	85	92	89	93	93
6 Std	61	88	42	89	30	78	34	92	91
6 Reg	32	92	34	93	27	63	21	87	89
7 Std	69	92	14	96	92	91	82	94	94
7 Reg	67	94	16	96	95	90	80	93	93
8 Std	27	93	10	97	81	92	73	93	95
8 Reg	41	95	7	95	89	88	76	94	93
Mean Std	60	93	36	95	80	91	77	94	94
Mean Reg	58	95	33	95	82	88	76	93	93

TABLE C.6: (continued). Parameters KPQ2–KYH.

ID	KPQ2	KQI	KQIVAR	KR0	KR1	KSR	KTW	KYH
1 Std	96	97	95	97	96	95	95	96
1 Reg	96	91	95	93	96	95	93	93
2 Std	92	95	95	96	94	93	94	94
2 Reg	96	95	97	92	95	94	91	92
3 Std	95	98	97	96	97	97	94	95
3 Reg	96	95	94	93	96	92	96	94
4 Std	97	91	97	94	95	90	87	95
4 Reg	97	89	97	95	96	93	95	94
5 Std	94	91	94	93	91	93	88	92
5 Reg	97	92	95	95	97	92	96	96
6 Std	91	82	89	87	91	87	87	91
6 Reg	95	76	89	75	88	83	76	87
7 Std	96	94	97	97	95	95	92	92
7 Reg	96	94	97	95	94	96	94	97
8 Std	94	92	94	94	93	88	91	93
8 Reg	94	93	94	90	94	90	86	90
Mean Std	94	93	95	94	94	92	91	93
Mean Reg	96	91	95	91	95	92	91	93

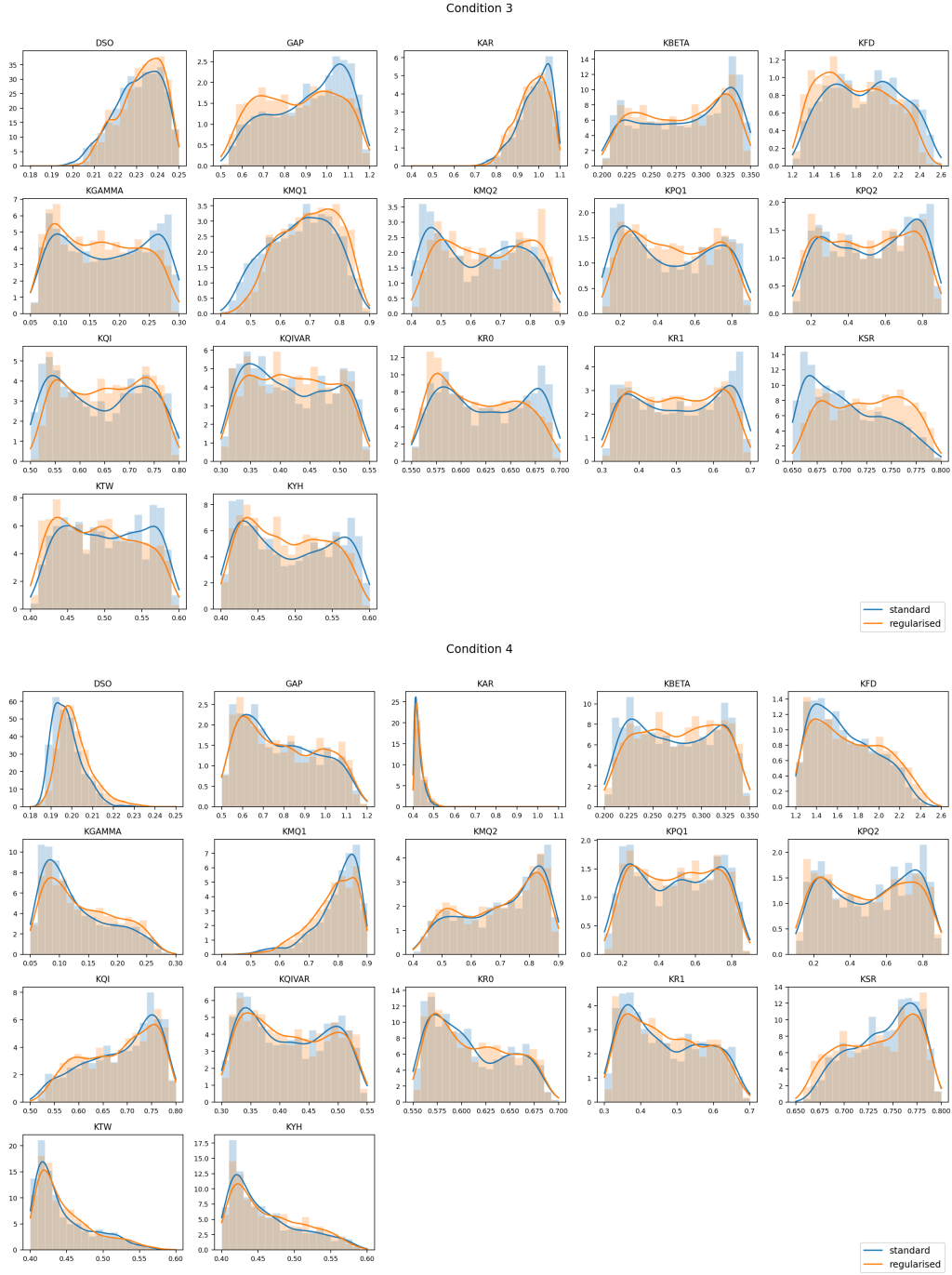


FIGURE C.15: Marginal distributions of the 1000 generated samples for the remaining 6 industrial conditions of §5.5.3, both model variants.

C. CVAE EVALUATION DETAILS

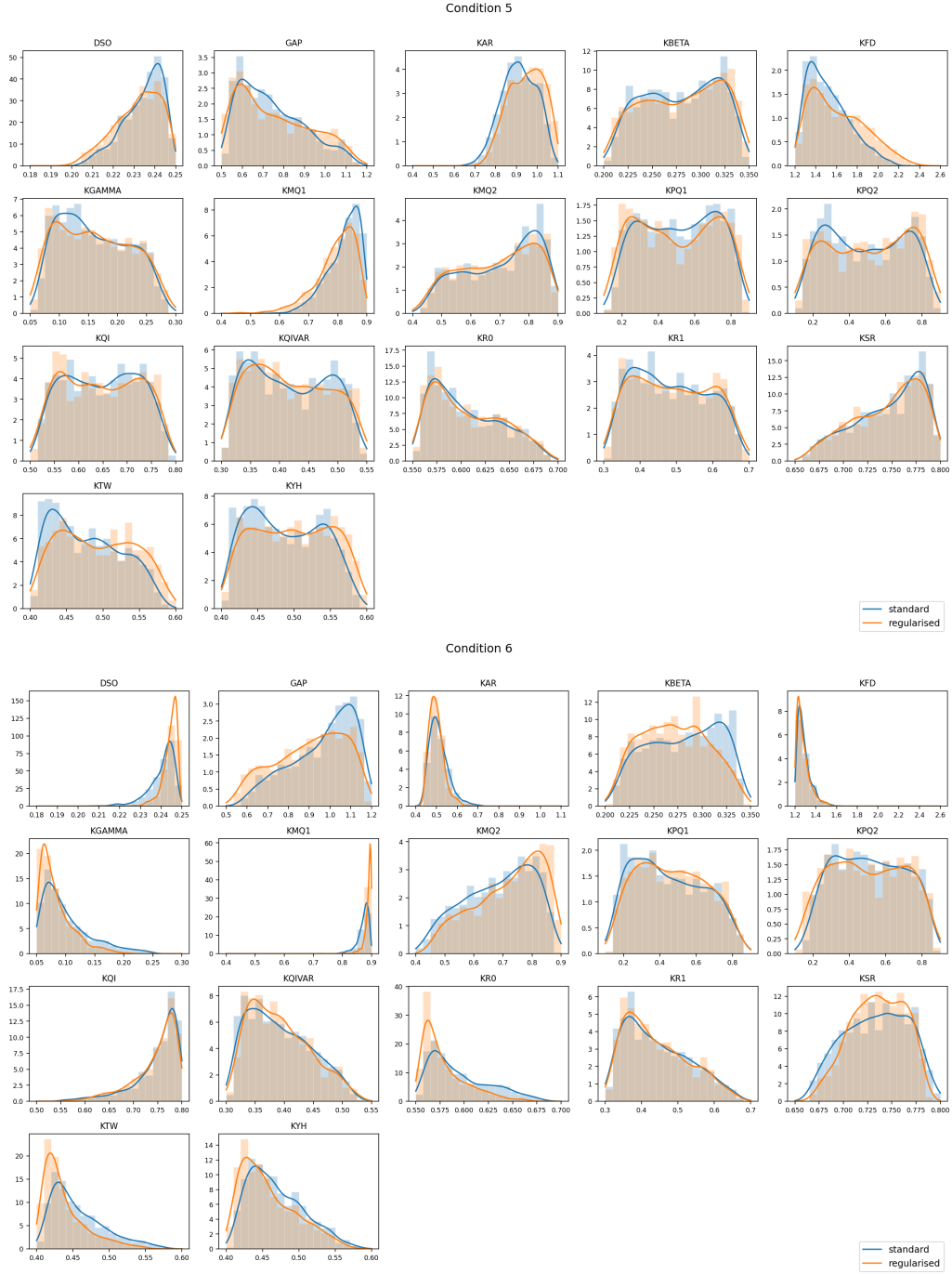


FIGURE C.15: (continued).

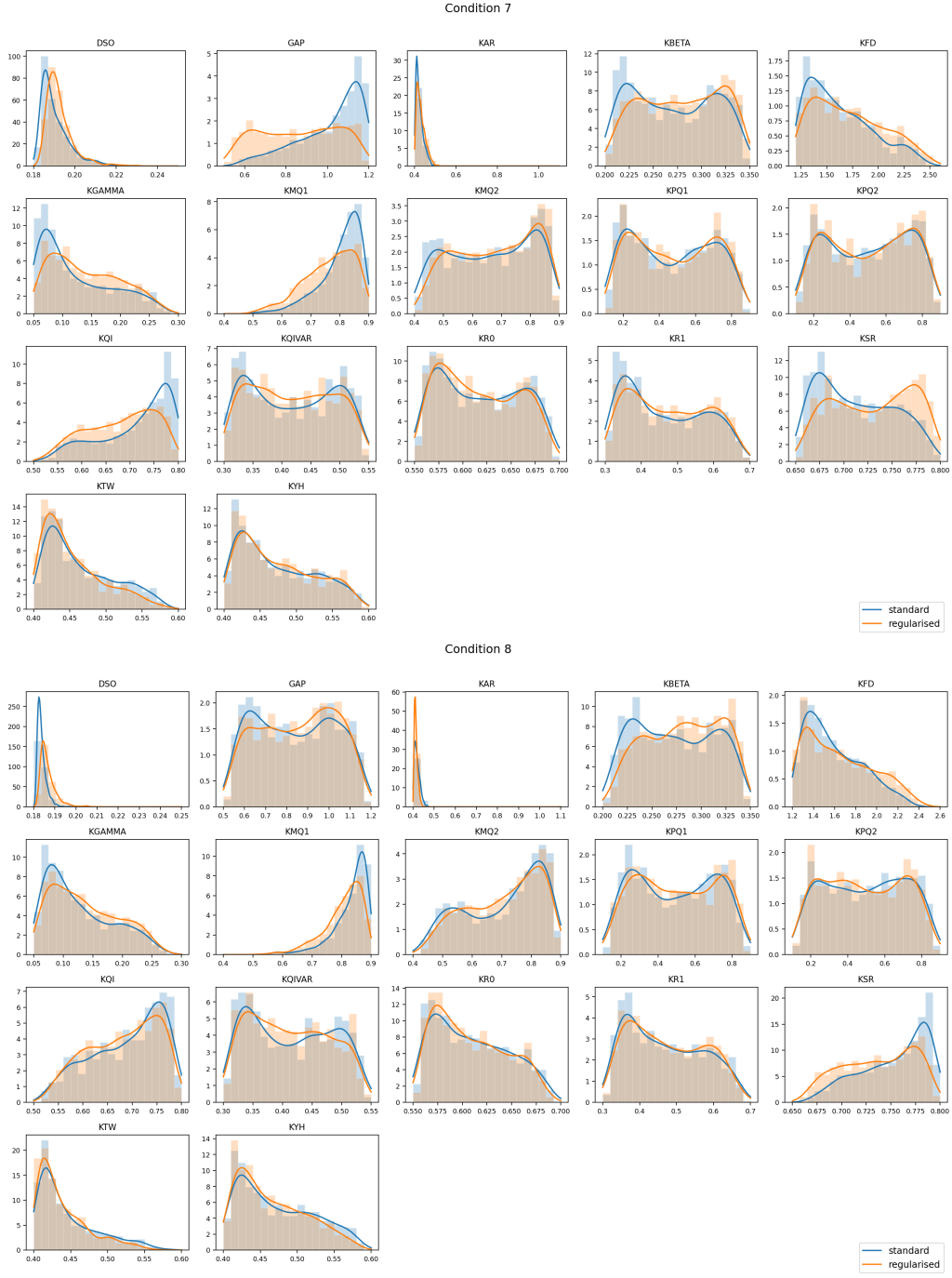


FIGURE C.15: (continued).

Appendix D

Latent-Space Optimisation Results

This appendix collects the optimisation results for Conditions 2–8 of Table 5.2; the panel layout and colour codes follow Figures 6.2–6.4 of §6.3. Per-condition convergence curves and baseline comparisons are reported first; the FEM validation of the winners is collected at the end.

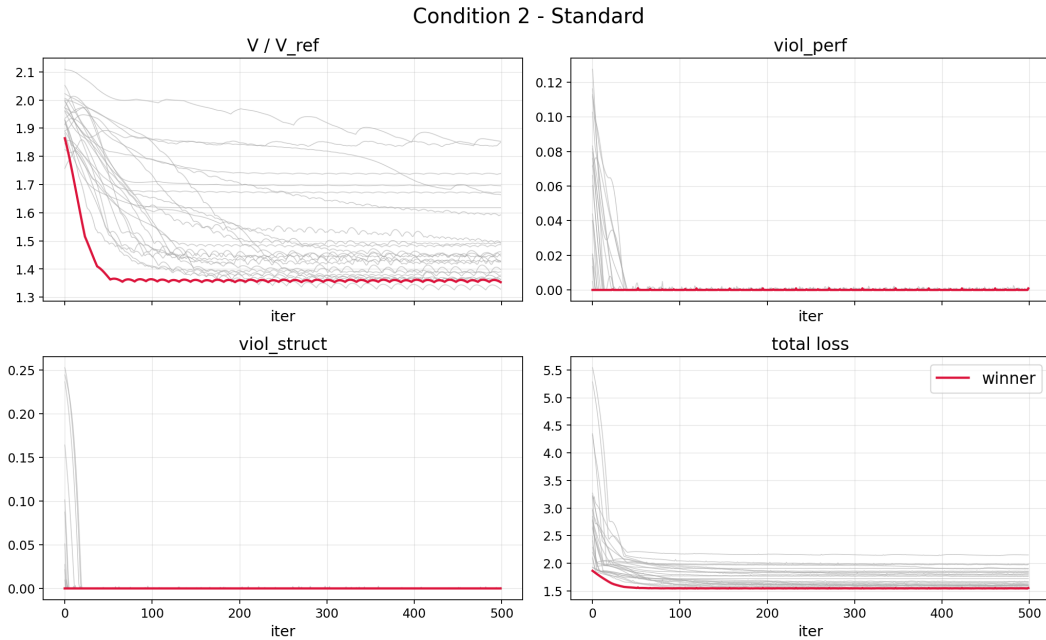


FIGURE D.1: Optimisation diagnostics for Condition 2. Convergence (top) and volume baseline comparison (bottom).

D. LATENT-SPACE OPTIMISATION RESULTS

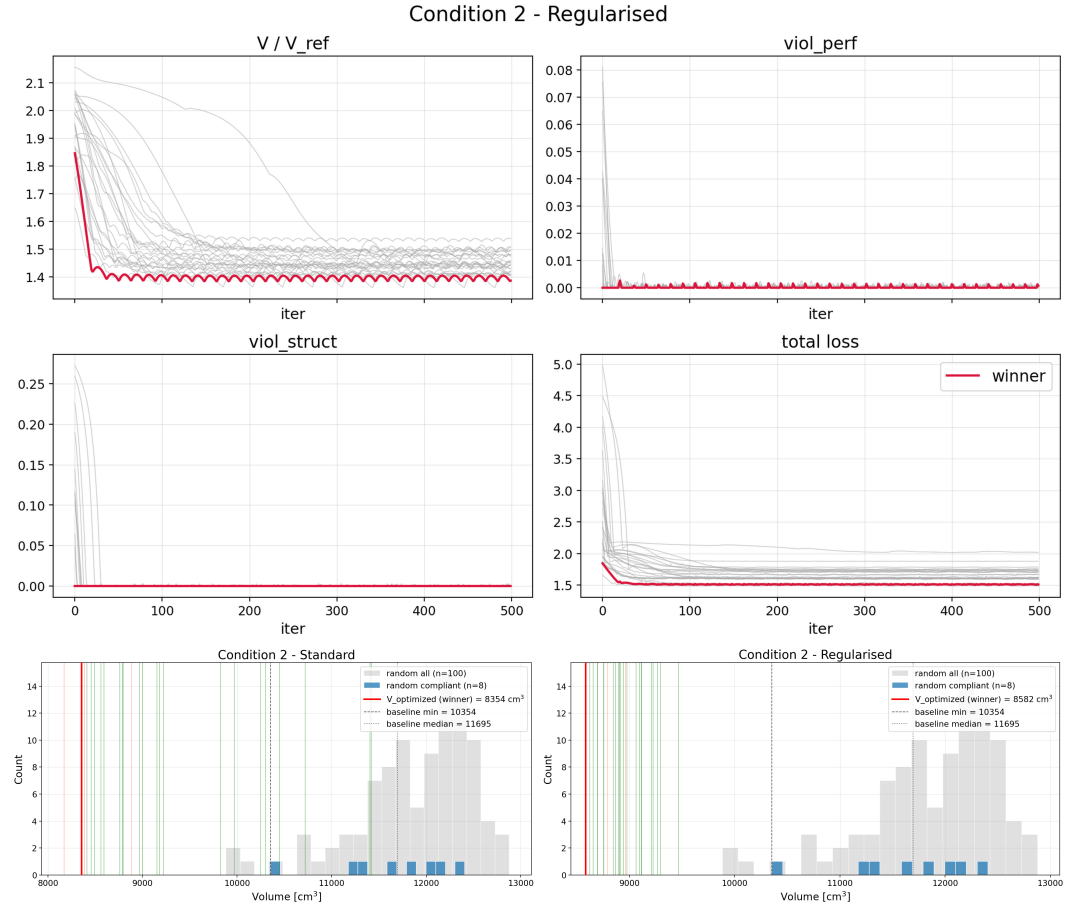


FIGURE D.1: (continued).

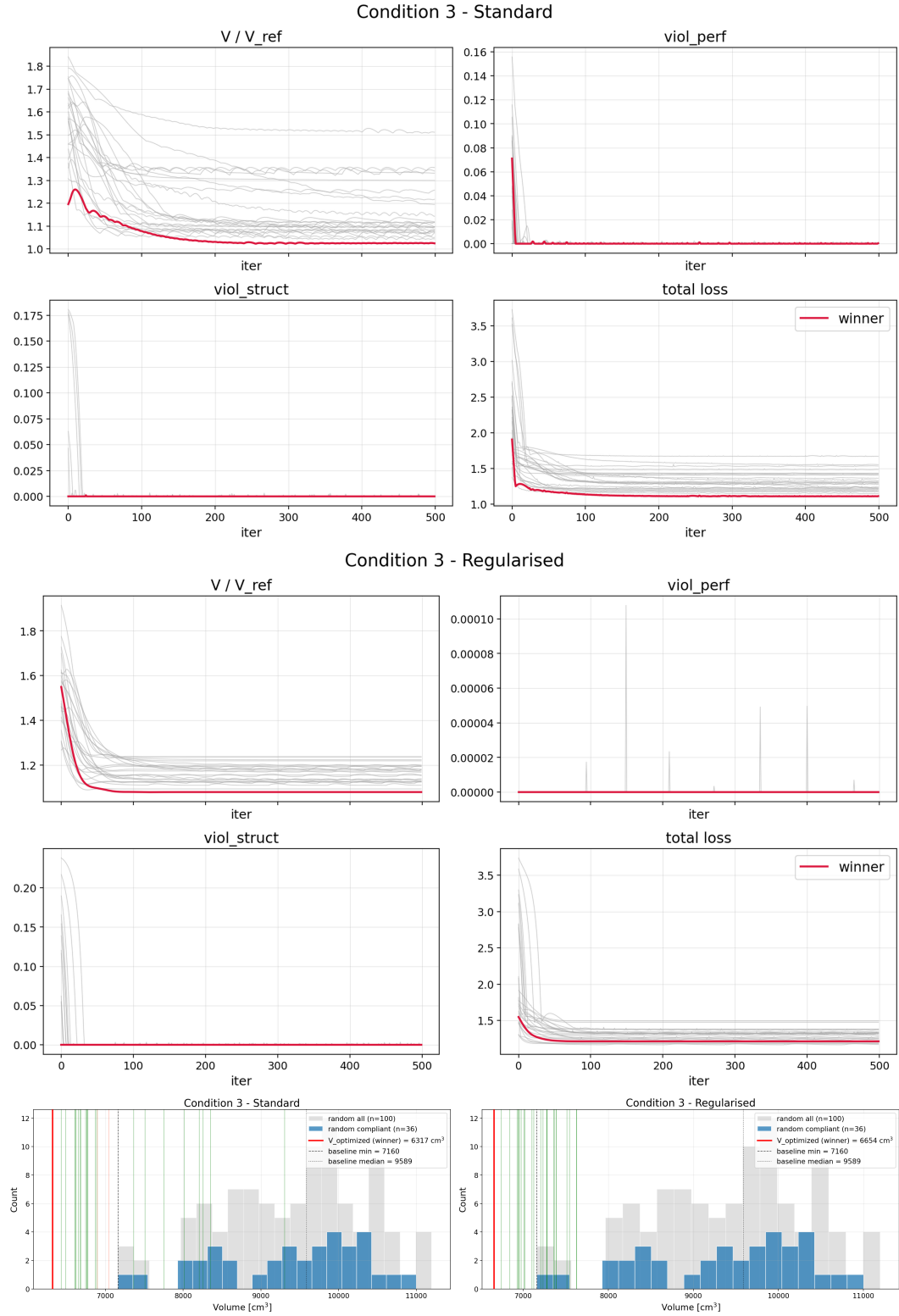


FIGURE D.2: Optimisation diagnostics for Condition 3. Convergence (top) and volume baseline comparison (bottom).

D. LATENT-SPACE OPTIMISATION RESULTS

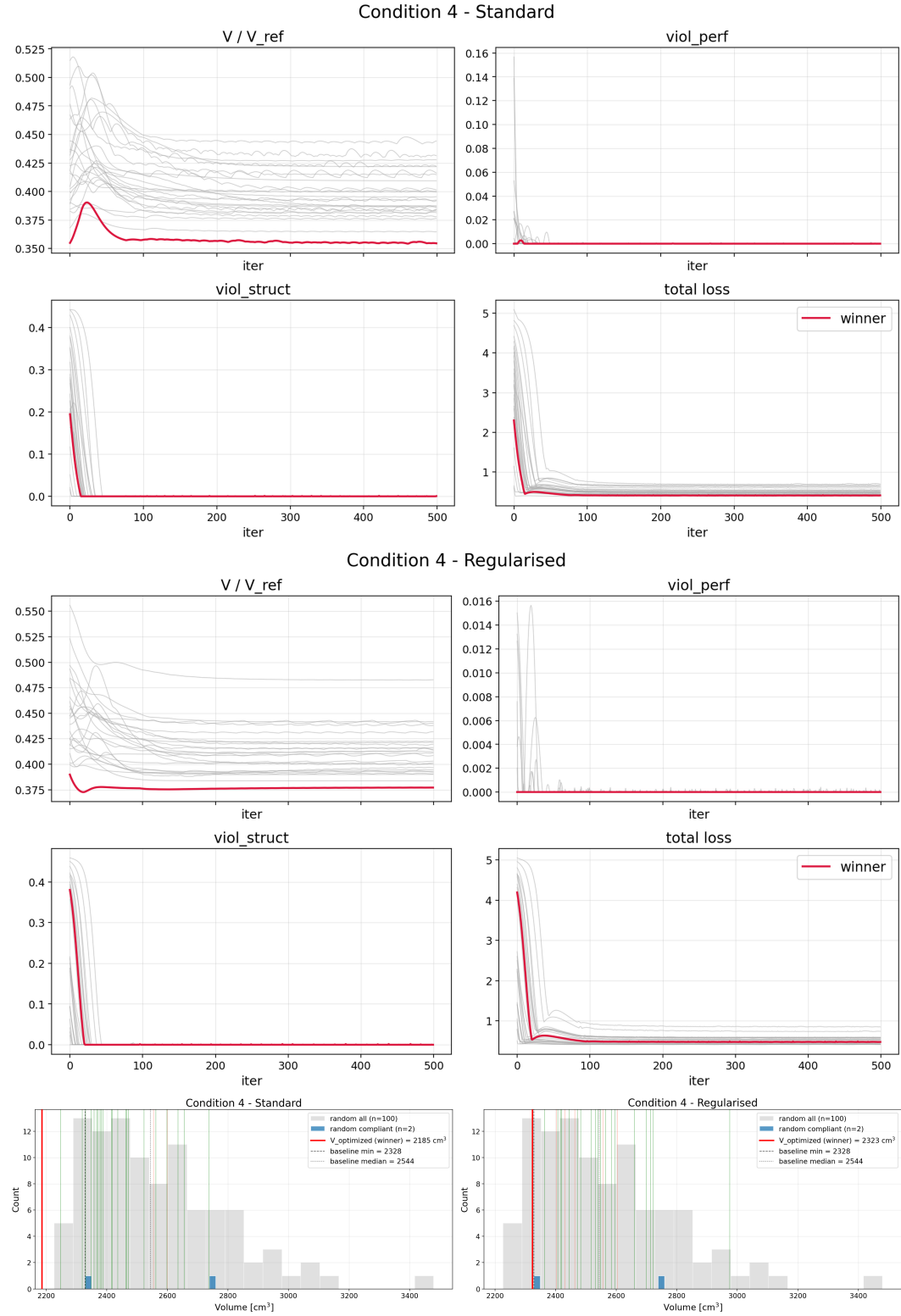


FIGURE D.3: Optimisation diagnostics for Condition 4. Convergence (top) and volume baseline comparison (bottom).

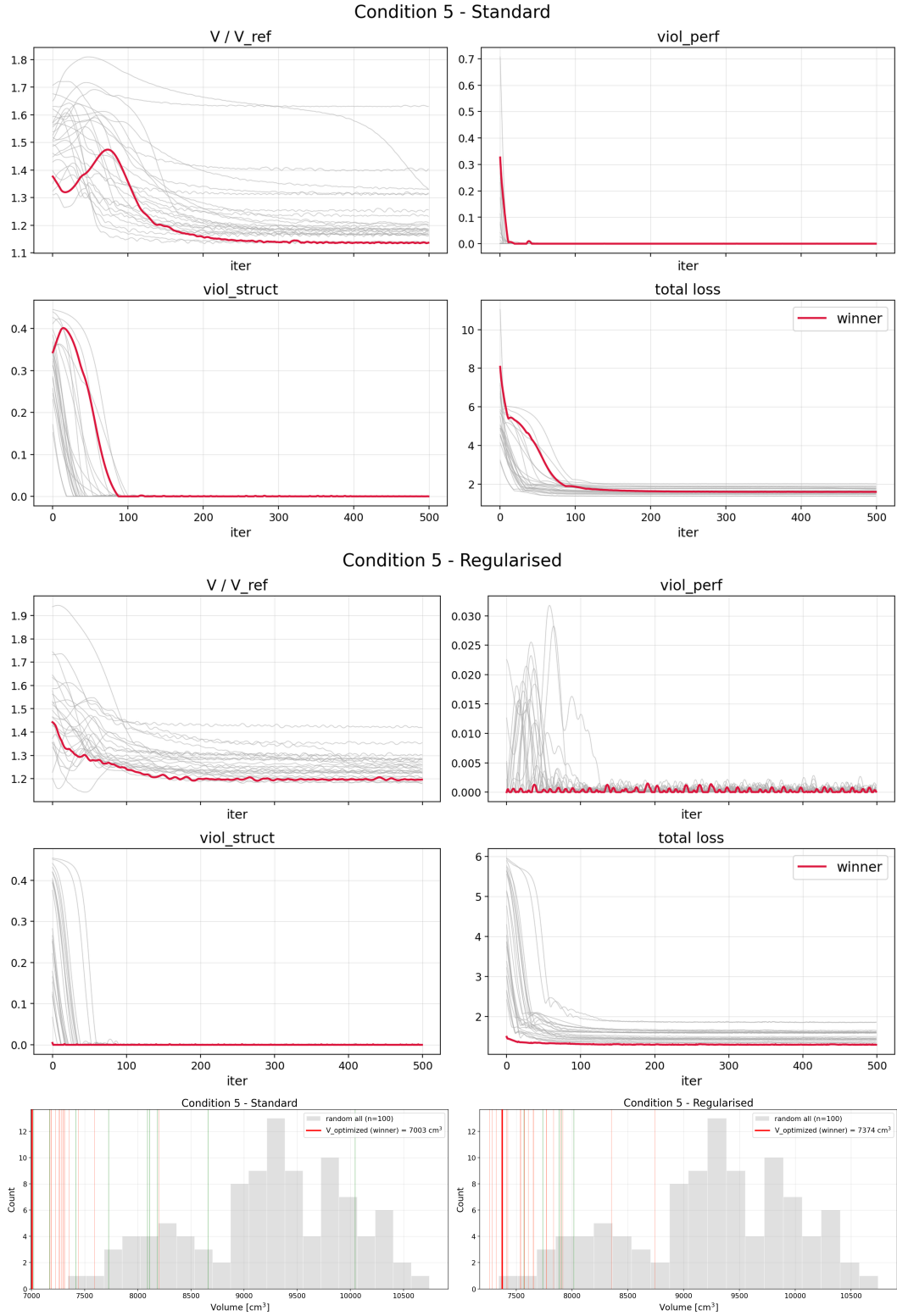


FIGURE D.4: Optimisation diagnostics for Condition 5. Convergence (top) and volume baseline comparison (bottom).

D. LATENT-SPACE OPTIMISATION RESULTS

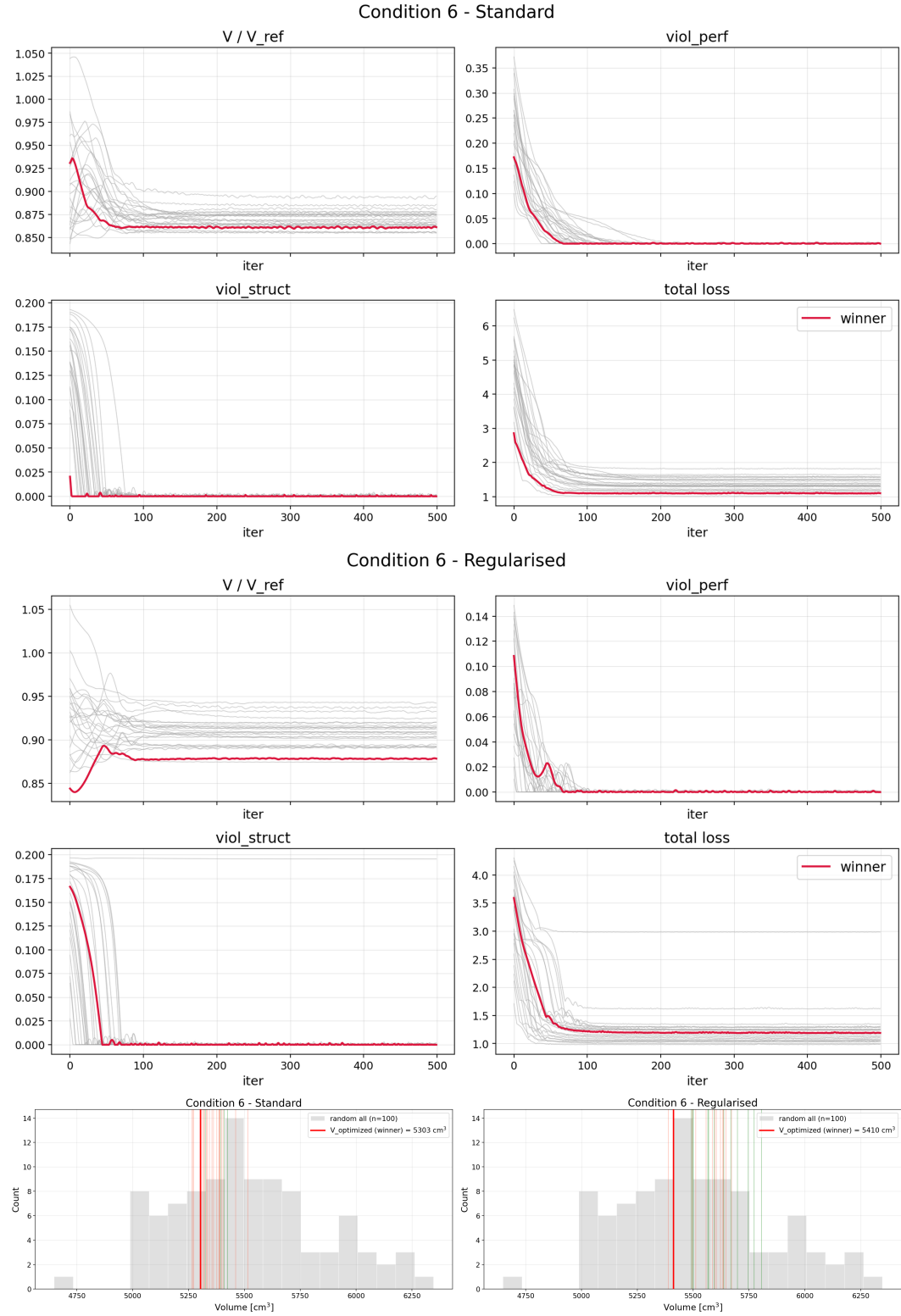


FIGURE D.5: Optimisation diagnostics for Condition 6. Convergence (top) and volume baseline comparison (bottom).

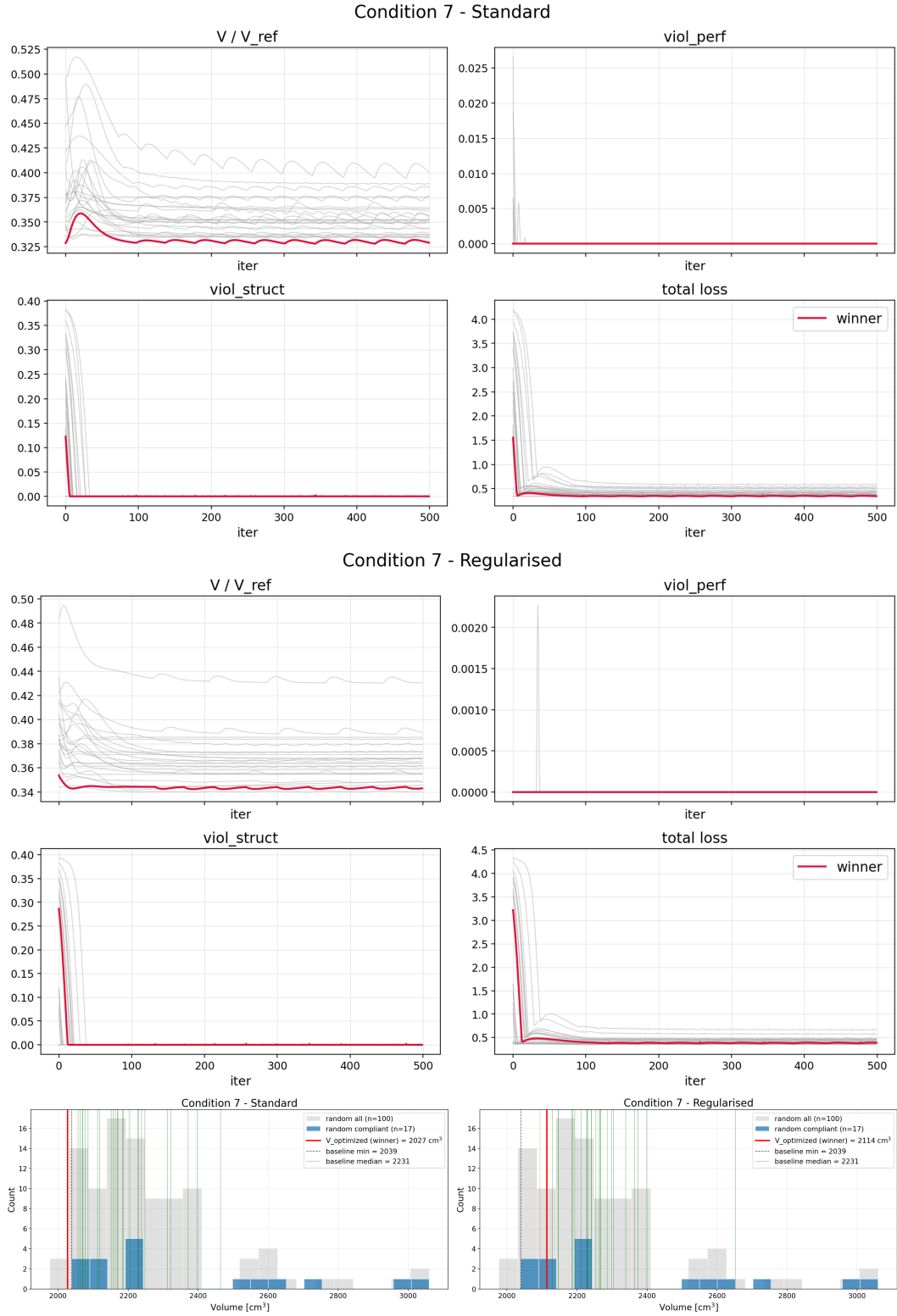


FIGURE D.6: Optimisation diagnostics for Condition 7. Convergence (top) and volume baseline comparison (bottom).

D. LATENT-SPACE OPTIMISATION RESULTS

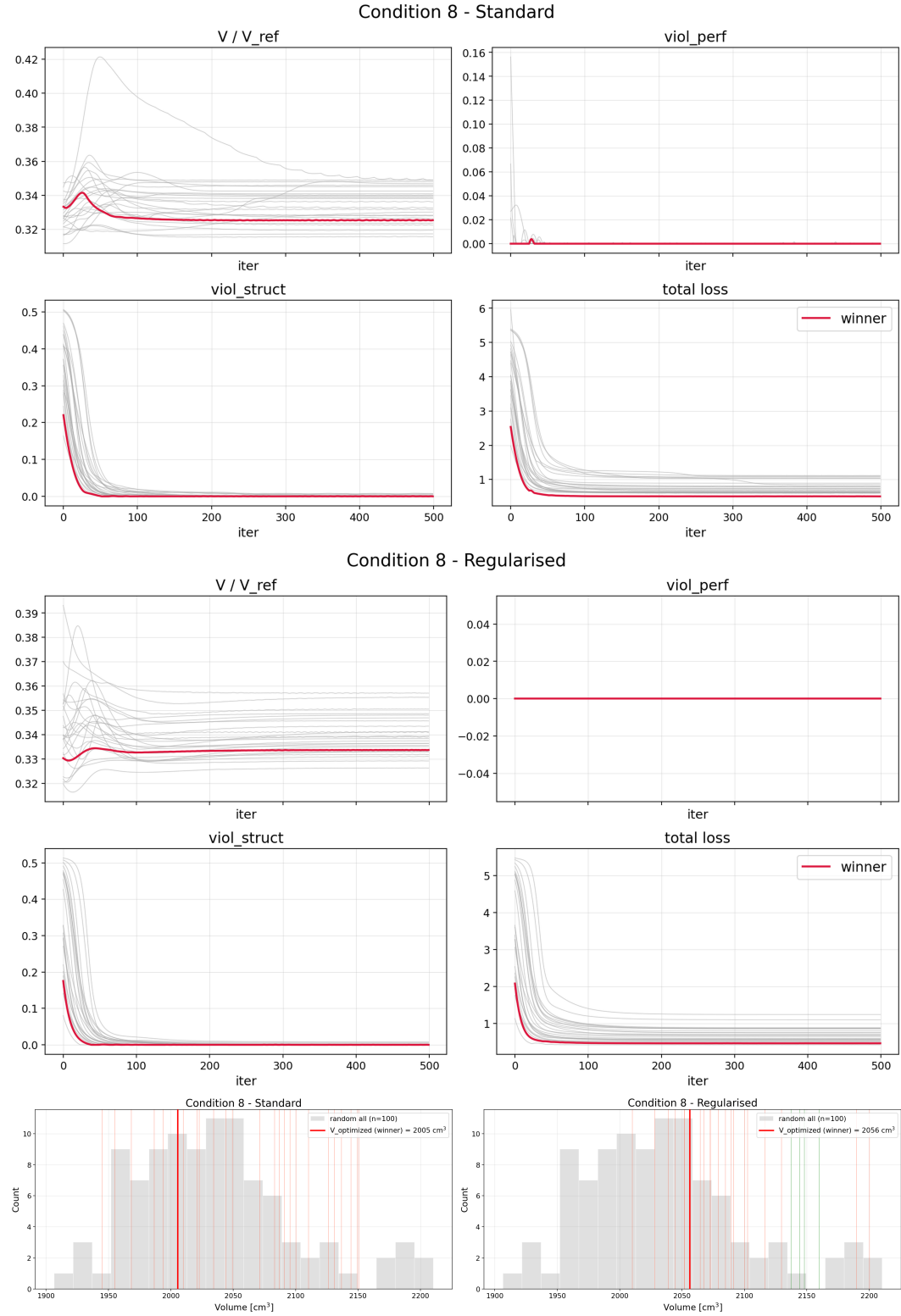


FIGURE D.7: Optimisation diagnostics for Condition 8. Convergence (top) and volume baseline comparison (bottom).

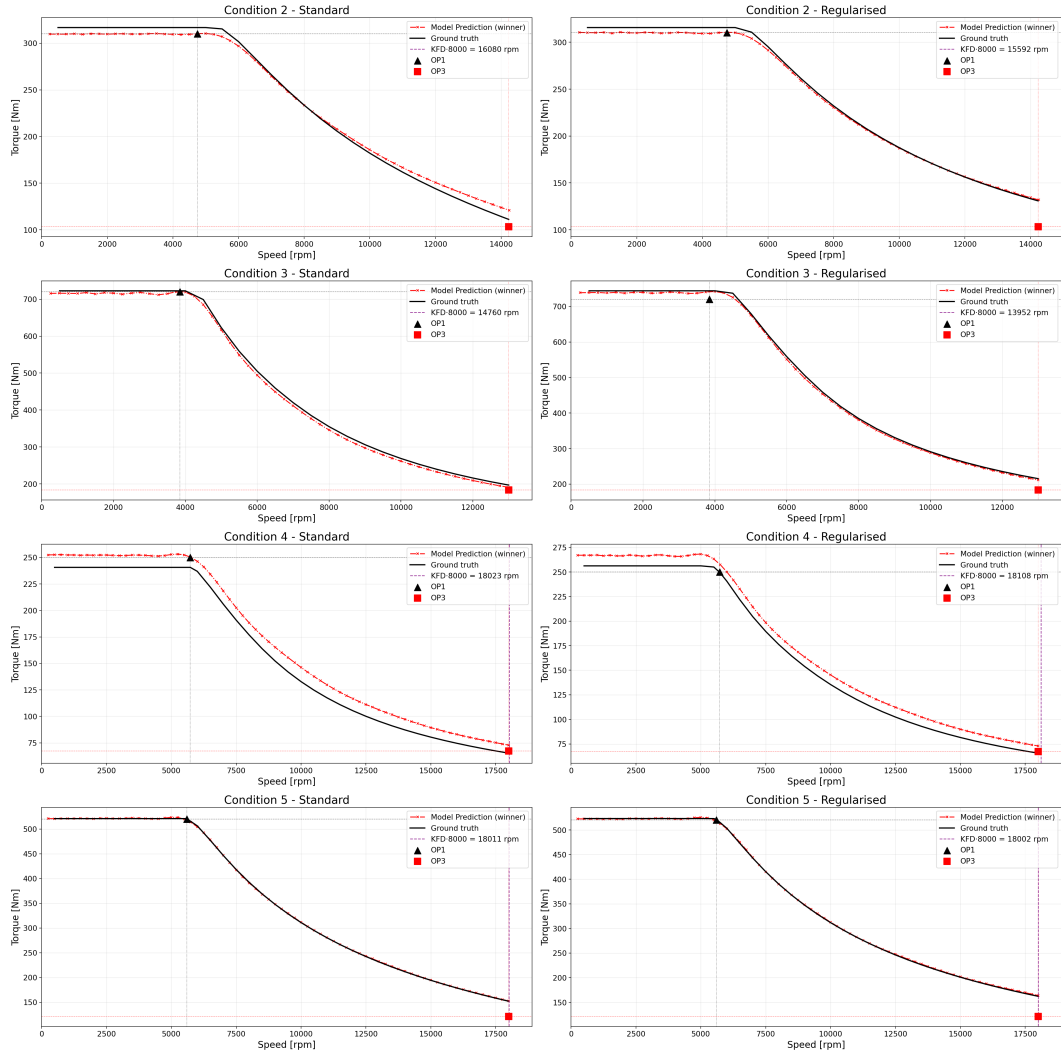


FIGURE D.8: FEM validation of the optimisation winners for Conditions 2–5: MLP-predicted vs FEM ground-truth torque–speed curves, with OP1 and OP3 marked. Standard (left) and Regularised (right).

Figure D.8 reveals one minor discrepancy: for Condition 4 the MLP-predicted torque sits slightly above the FEM ground truth, so the surrogate-based winner-selection rule reports a compliant design while the FEM-evaluated torque falls marginally short of the requested T_{OP_i} . A practical mitigation could be to increase the requested T_{OP1} and T_{OP3} by a small margin for that condition: the decoder then biases the generated geometry towards higher predicted torque, restoring an FEM-compliant winner without changing the optimiser itself.

D. LATENT-SPACE OPTIMISATION RESULTS

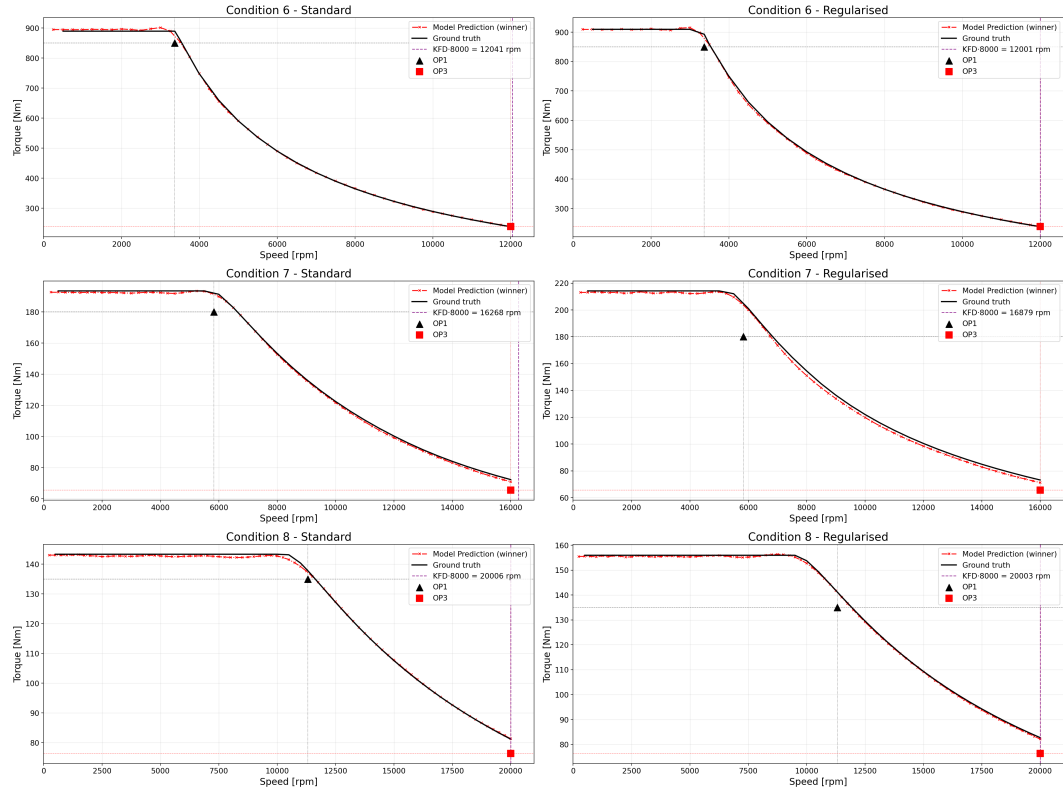


FIGURE D.9: FEM validation of the optimisation winners for Conditions 6–8.

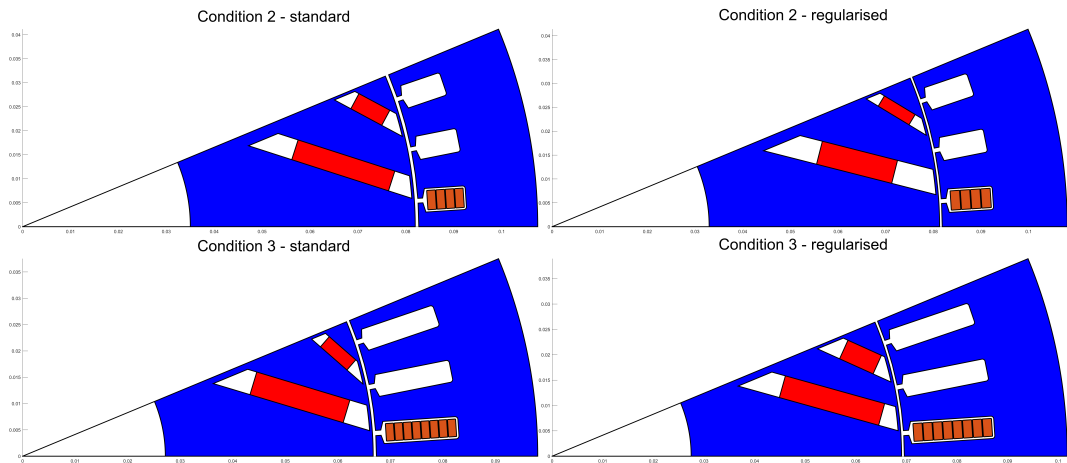


FIGURE D.10: One-pole slice of the winning geometries for Conditions 2–8. Standard (left) and Regularised (right) CVAE.

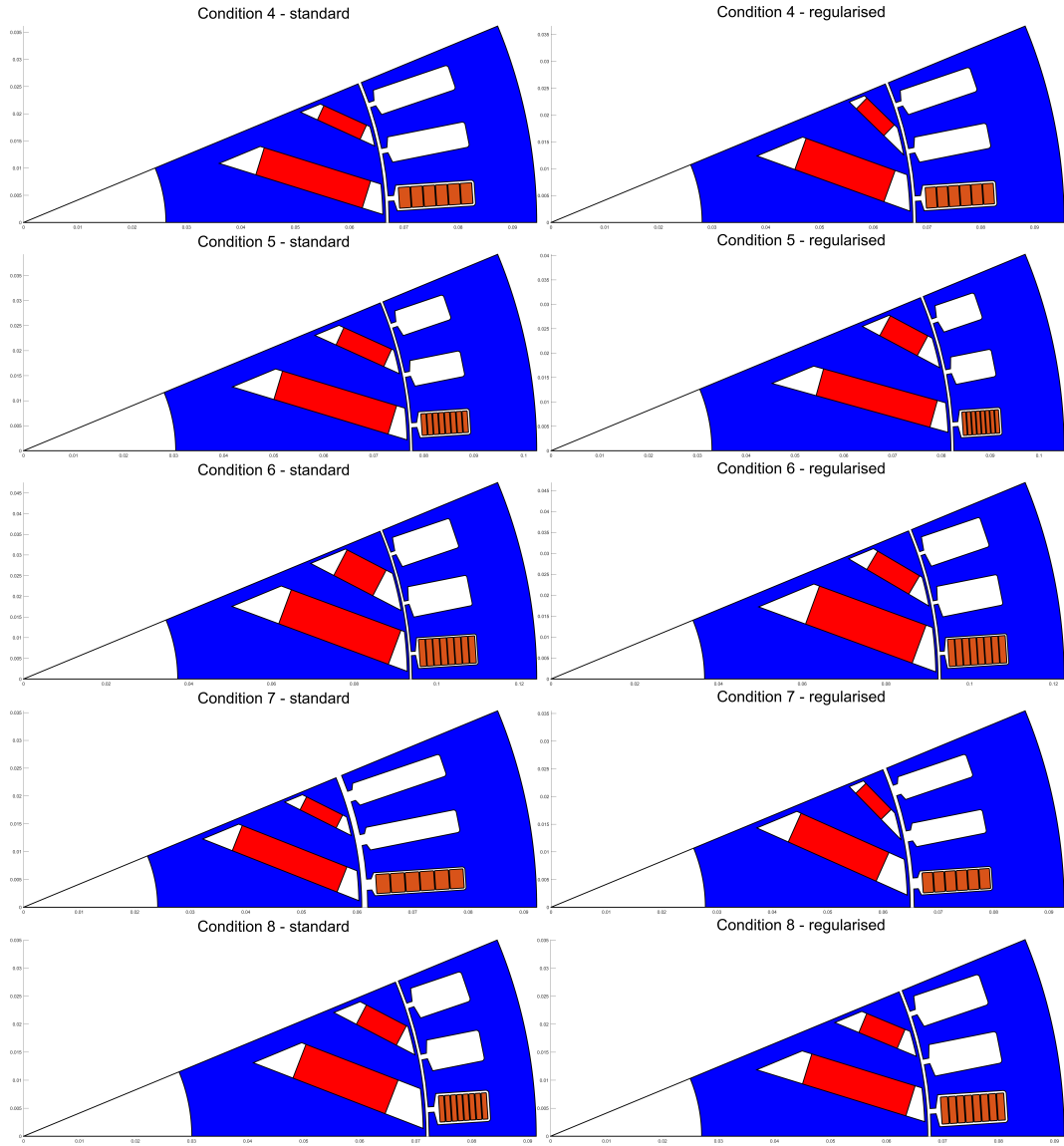


FIGURE D.10: (continued).

Bibliography

- [1] A. Akbari and D. A. Lowther. CDC-GANs: Bridging Innovation and Efficiency in E-Machine Design with Advanced Generative Models. In *2024 International Conference on Electrical Machines (ICEM)*, pages 1–7, Torino, Italy, Sept. 2024. IEEE.
- [2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.
- [3] S. Barmada, P. Di Barba, N. Fontana, M. E. Mognaschi, and M. Tucci. Electromagnetic Field Reconstruction and Source Identification Using Conditional Variational Autoencoder and CNN. *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 8:322–331, 2023.
- [4] B. Bilgin, P. Magne, P. Malysz, Y. Yang, V. Pantelic, M. Preindl, A. Korobkine, W. Jiang, M. Lawford, and A. Emadi. Making the Case for Electrified Transportation. *IEEE Transactions on Transportation Electrification*, 1(1):4–17, June 2015.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.
- [6] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 10–21, Berlin, Germany, 2016. Association for Computational Linguistics.
- [7] W. Cai, X. Wu, M. Zhou, Y. Liang, and Y. Wang. Review and Development of Electric Motor Systems and Electric Powertrains for New Energy Vehicles. *Automotive Innovation*, 4(1):3–22, Feb. 2021.
- [8] S. De Vos. Flux map prediction of IPMSM with a neural network based approach. Master’s thesis, KU Leuven. Faculteit Ingenieurswetenschappen, 2025.
- [9] P. Di Barba, M. E. Mognaschi, and S. Wiak. Neural metamodelling of fields: Towards a new deal in computational electromagnetics. *International Journal of Applied Electromagnetics and Mechanics*, 69(2):127–137, June 2022.
- [10] D. Di Lorenzo, V. Champaney, C. Ghnatios, E. Cueto, and F. Chinesta. Physics-informed and graph neural networks for enhanced inverse analysis. *Engineering Computations*, 42(7):2427–2455, Nov. 2025.

- [11] C. P. Do Nascimento, A. C. Silvério, B. Baptista, J. A. Carvalho, T. Bazzo, and Á. Ferreira. Optimization approaches in electric machine design: insights from a bibliometric analysis. *Journal of Engineering and Applied Science*, 73(1):2, Dec. 2026.
- [12] S. Fang and H. Ji. ICVAE: Interpretable Conditional Variational Autoencoder for de novo Molecular Design. *International Journal of Molecular Sciences*, 26(9):3980, 2025.
- [13] C. Fu, H. Kazmi, M. Quintana, and C. Miller. Creating synthetic energy meter data using conditional diffusion and building metadata. *Energy and Buildings*, 312:114216, June 2024.
- [14] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.
- [15] J. Hauke and T. Kossowski. Comparison of values of Pearson’s and Spearman’s correlation coefficients on the same sets of data. *Quaestiones Geographicae*, 30(2):87–93, 2011.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [17] D. Hendrycks and K. Gimpel. Gaussian Error Linear Units (GELUs), 2016.
- [18] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.
- [19] D. G. Holmes and T. A. Lipo. *Pulse Width Modulation for Power Converters: Principles and Practice*. IEEE Press Series on Power Engineering. Wiley-IEEE Press, Piscataway, NJ, 2003.
- [20] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [21] P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- [22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
- [23] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

-
- [24] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.
 - [25] D. P. Kingma and M. Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, Nov. 2019.
 - [26] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff. *Analysis of Electric Machinery and Drive Systems*. Wiley-IEEE Press, Hoboken, NJ, 3 edition, 2013.
 - [27] S. Lauwers. Use of neural networks in electrical machine design. Master’s thesis, KU Leuven. Faculteit Ingenieurswetenschappen, 2024.
 - [28] C. Liu, K. T. Chau, C. H. T. Lee, and Z. Song. A Critical Review of Advanced Electric Machines and Control Strategies for Electric Vehicles. *Proceedings of the IEEE*, 109(6):1004–1028, June 2021.
 - [29] G. Loaiza-Ganem and J. P. Cunningham. The continuous Bernoulli: fixing a pervasive error in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.
 - [30] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.
 - [31] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.
 - [32] MathWorks. Clarke and Park Transforms. <https://nl.mathworks.com/discovery/clarke-and-park-transforms.html>, 2026. Accessed 2026-04-27.
 - [33] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
 - [34] M. H. Mohammadi. *A Data-Driven Design Process including Multiphysics for Synchronous AC Machines using High-Performance Computing*. PhD thesis, McGill University, 2019.
 - [35] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.
 - [36] H. T. Nguyen and P. L. Donti. FSNet: Feasibility-Seeking Neural Network for Constrained Optimization with Guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
 - [37] V. Parekh, D. Flore, and S. Schöps. Deep learning based Meta-modeling for Multi-objective Technology Optimization of Electrical Machines. *IEEE Access*, 11:93420–93430, 2023.

- [38] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1310–1318, 2013.
- [39] J. Pyrhönen, T. Jokinen, V. Hrabovcová, and H. Niemelä. *Design of rotating electrical machines*. Wiley, Chichester, reprinted edition, 2010.
- [40] PyTorch Contributors. BCEWithLogitsLoss — PyTorch documentation. <https://docs.pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html>, 2026. Accessed 2026-05-20.
- [41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [42] L. Ruthotto and E. Haber. An Introduction to Deep Generative Modeling. *GAMM-Mitteilungen*, 44(2):e202100008, 2021.
- [43] W. Soong and T. Miller. Field-weakening performance of brushless synchronous AC motor drives. *IEE Proceedings - Electric Power Applications*, 141(6):331–340, Nov. 1994.
- [44] S. Stipetic, D. Zarko, and M. Popescu. Scaling laws for synchronous permanent magnet machines. In *2015 Tenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, pages 1–7, Monte Carlo, Mar. 2015. IEEE.
- [45] M. Tanveer, M. M. Azad, D. Kim, S. Khalid, and H. S. Kim. Generative Design for Engineering Applications: A State-of-the-Art Review. *Archives of Computational Methods in Engineering*, 33(1):53–79, Jan. 2026.
- [46] J. M. Tomczak. *Deep Generative Modeling*. Springer, Cham, 2 edition, 2024.
- [47] S. Yang, S. Lee, and K. Yee. Inverse design optimization framework via a two-step deep learning approach: application to a wind turbine airfoil. *Engineering with Computers*, 39(3):2239–2255, June 2023.
- [48] Y. Yang, S. M. Castano, R. Yang, M. Kasprzak, B. Bilgin, A. Sathyan, H. Dadkhah, and A. Emadi. Design and Comparison of Interior Permanent Magnet Motor Topologies for Traction Applications. *IEEE Transactions on Transportation Electrification*, 3(1):86–97, Mar. 2017.