

WHICH CONSTRAINTS ARE MISSING? ASK THE VERIFIER: GRADED REWARDS FOR CONSTRAINT-FOLLOWING MUSIC GENERATION

Haoyue Liu^{1,3} Xiaoyu Ma¹ Ye Chen² Zhichao Wang¹ Haoran Shou¹ Xiaoying Tang^{1,3,†}

¹ School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China

² XJTU-POLIMI Joint School, Xi'an Jiaotong University, Xi'an 710049, China

³ Shenzhen Future Network of Intelligence Institute (FNii-Shenzhen)

ABSTRACT

Existing work on symbolic music generation has focused on musicality and perceptual quality, while paying less attention to whether models can jointly satisfy user-specified score constraints. Existing evaluations also lack a dedicated benchmark for this capability. Accordingly, we construct **MUSICCONSTRAINTBENCH**, a benchmark of 2,180 items spanning eight families of programmatically verifiable constraints, and find that joint satisfaction drops sharply as constraints accumulate. A natural solution is to use these verifiers as reinforcement-learning rewards, but binary all-satisfied rewards provide sparse supervision: during the first 50 updates, 0.550 of rollout groups receive identical rewards and thus no reward gradient; meanwhile, such a binary criterion cannot distinguish partially compliant outputs from complete failures. To address this limitation, we introduce **MUSICRLVR**, a verifier-driven reinforcement learning framework that combines a hard validation gate, graded per-property credit, and a joint-satisfaction bonus, requiring no human annotation, learned reward model, or music-domain SFT. On **MUSICCONSTRAINTBENCH**, **MUSICRLVR** improves Qwen3-4B-Instruct from 0.160 to 0.807 in joint satisfaction on mixed constraints, enabling the 4B model to outperform Llama-3.1-70B. It generalises to property combinations unseen during training and to out-of-range parameter values, showing that verifiable rewards can train open-ended generation models without presupposing a unique target output.

1 INTRODUCTION

Language models can now generate machine-readable symbolic scores from free-text descriptions (Yuan et al., 2024; Qu et al., 2025; Wang et al., 2025a). For users who want a piece that satisfies explicit requirements rather than an arbitrary piece, however, a more natural interface is a score specification that names properties such as key, meter, length, pitch range, final note, rhythmic vocabulary, melodic motion, and form. Unlike perceptual quality (Cideron et al., 2024; Jonason et al., 2025; Ziv et al., 2026), these properties can be evaluated programmatically from the parsed score, placing constraint-following music generation in a class of *property-verifiable generation* problems. Yet existing evaluations do not systematically measure whether models can satisfy such constraints jointly, as no dedicated benchmark isolates this capability. We therefore construct **MUSICCONSTRAINTBENCH**, a benchmark of 2,180 ABC-generation items spanning eight constraint families (Yuan et al., 2024; Qu et al., 2025; Wang et al., 2025a). With this benchmark, we ask a central question: when each requested property is individually verifiable, can a model actually satisfy them jointly in a single generation?

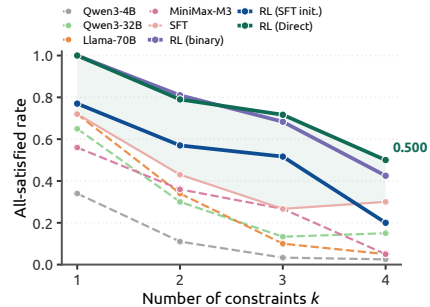


Figure 1: All-satisfied rate against k on Mixed, for the matched runs and the baselines of Table 2; each slice has its own family mixture. The right-hand label gives the Direct MusicRLVR rate at $k = 4$.

Our experiments reveal a sharp collapse in multi-constraint compliance. A request may simultaneously require a tune to be *in D minor, in 6/8, sixteen bars long, ending on the tonic, with limited melodic leaps and an AABA form*; because these requirements must hold jointly, violating any one of them makes the output non-compliant. As the number of simultaneous constraints increases, the all-satisfied rate drops sharply (Figure 1). A natural solution is to use the deterministic verifiers directly as reinforcement-learning rewards, but a binary all-satisfied reward inherits the same conjunction problem: once any constraint fails, outputs with very different degrees of partial compliance receive the same reward. In GRPO, this directly removes the learning signal when all rollouts in a group receive the same reward. In our matched training runs, 0.550 of rollout groups have zero reward variance over the first 50 updates and therefore provide no reward-driven gradient. Thus, while a verifier can determine whether an output is fully correct, a binary verdict cannot distinguish partially compliant outputs from complete failures. This motivates our core question: *Can deterministic property checks provide the task-specific supervision a general-purpose model needs to jointly follow multiple musical instructions?*

Turning these verifiers into a training reward therefore requires more than a simple pass/fail signal. Yet assigning partial credit alone creates the opposite problem: a fragment may satisfy several local properties and receive reward without forming a valid tune. To address this, we introduce MUSICRLVR, a verifier-driven reinforcement learning framework with two complementary components. First, a hard validation gate rejects malformed, unparseable, or otherwise invalid score outputs before any property credit is assigned. Second, for outputs that pass the gate, an item-specific graded reward assigns per-property credit for the constraints requested by the prompt and adds a joint-satisfaction bonus when all requirements are met. For the more structured form constraint, we additionally provide graded feedback within the constraint family itself, allowing training to distinguish different degrees of partial success. The framework requires no human annotation or learned reward model.

We summarise our contributions as follows:

- **The first benchmark for joint score-constraint compliance.** We introduce MUSICCONSTRAINT-BENCH, to our knowledge the first benchmark specifically designed to evaluate whether language models can jointly satisfy multiple programmatically verifiable score constraints. It contains 2,180 items spanning eight verifier-defined constraint families, plus 400 paraphrases, and systematically tests constraint density, unseen family combinations, parameter extrapolation, and wording robustness.
- **Turning property verifiers into fine-grained training rewards.** We introduce MUSICRLVR, which combines a hard validity gate, graded per-family credit, and a joint-satisfaction bonus without human annotation or a learned reward model. The verifier configuration is item-specific, and the reward provides fine-grained supervision both across requested constraint families and, for form, within a single family.
- **Training constraint following without music-domain SFT.** Direct GRPO from Qwen3-4B-Instruct raises the Mixed joint-satisfaction rate from 0.160 to 0.807, enabling the 4B model to outperform every evaluated zero-shot baseline, including Llama-3.1-70B. The gains further generalise to property combinations never observed together during training and to parameter values outside the training range.

2 RELATED WORK

Symbolic music language models, from LSTMs over ABC (Sturm et al., 2016) and the Music Transformer (Huang et al., 2018) to ABC-native LLMs (Yuan et al., 2024; Qu et al., 2025; Wang et al., 2025a), description-conditioned control (von Rütte et al., 2023; Thickstun et al., 2023) and audio models (Agostinelli et al., 2023; Copet et al., 2023), are trained by likelihood or against a learned music-text encoder (Wu et al., 2025); hard rules enter, if at all, at decoding time (Lu et al., 2021; Huang et al., 2024; Kaliakatsos-Papakostas et al., 2025). RL has been applied in two directions: optimising a human or learned quality signal (Cideron et al., 2024; Jonason et al., 2025; Ziv et al., 2026; He et al., 2026), or rewarding music-theory heuristics as a style prior (Jaques et al., 2017). However, all of these score with a human, a learned model, or a fixed prior, none of which can report whether the properties *this* prompt requested actually hold. In contrast, we score with a deterministic checker whose family and parameters change from item to item, and test combinations withheld from training.

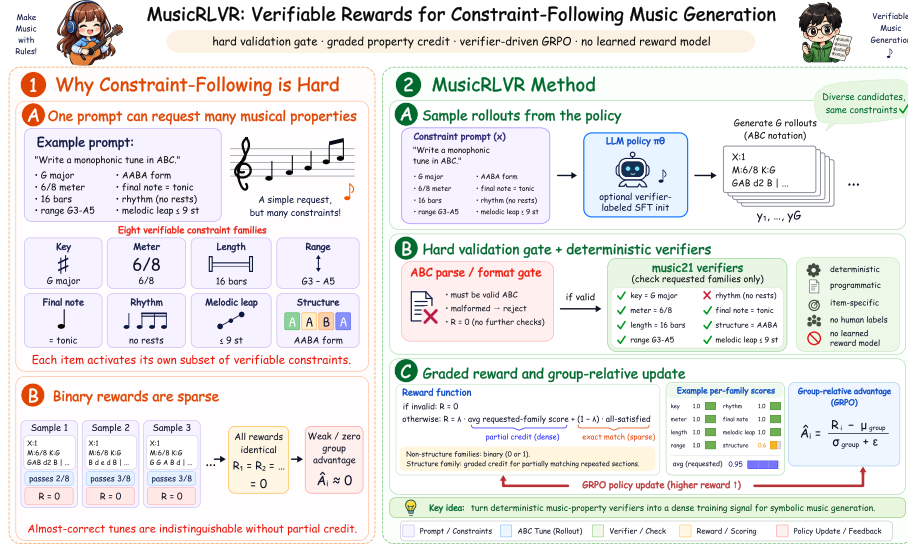


Figure 2: **Overview of MusicRLVR.** **Left:** Multi-constraint prompts make binary all-satisfied rewards sparse, leaving partially correct rollouts indistinguishable. **Right:** MusicRLVR samples G ABC rollouts, applies a hard validation gate and deterministic property verifiers, and uses graded per-family credit plus an all-satisfied bonus to provide the group-relative signal for GRPO.

RLVR grew out of PPO-based RLHF (Schulman et al., 2017; Ouyang et al., 2022), with group-relative baselines (Ahmadian et al., 2024; Shao et al., 2024) removing the value network; it underpins recent reasoning models (Guo et al., 2025; Lambert et al., 2024) and is being pushed beyond mathematics (Ma et al., 2026; Gunjal et al., 2026). DAPO (Yu et al., 2026) and Dr. GRPO (Liu et al., 2025) patch dead groups and length bias, whereas we grade the reward itself (§3.3); like IFEval (Zhou et al., 2023) we score only verifiable instructions. Closest in domain, Wang et al. (2025b) synthesise verifiable sheet-music reasoning questions and study transfer to symbolic music continuation. Our task instead requires joint satisfaction of multiple musical properties for which no target composition exists: the reward is computed from the properties the generation prompt requests, and we evaluate held-out combinations and parameter extrapolation rather than reasoning-question accuracy.

3 METHOD

MusicRLVR trains a policy to satisfy the properties requested by each prompt using a hard validation gate and graded property-level rewards (Figure 2). Algorithm 1 applies this reward in GRPO, optionally after verifier-labelled SFT.

3.1 PROBLEM FORMULATION

Let $\mathcal{F}$ be a set of *constraint families*; we use the eight families in Table 1. A *constraint* $c = (f, \theta)$ pairs a family $f \in \mathcal{F}$ with parameters θ (e.g. $f = \text{key}$, $\theta = (\text{G}, \text{major})$). Each family has a deterministic *verifier* $v_f: \mathcal{S} \times \Theta_f \rightarrow \{0, 1\}$, where $\mathcal{S}$ is the set of parseable scores and Θ_f the parameter space of family f . A *problem* is a set of constraints $C = \{c_1, \dots, c_k\}$ with distinct families, rendered into a prompt x_C by a fixed template. The policy π_ϕ , with trainable parameters ϕ , emits a string y in ABC text notation (Appendices C.5–C.7); ρ extracts the ABC and applies the validation gate of Appendix C.1, returning $\rho(y) \in \mathcal{S}$ or $\perp$ on rejection. The training reward uses the binary indicator

$$A(y, C) = \prod_{(f, \theta) \in C} v_f(\rho(y), \theta) \in \{0, 1\} \quad \text{if } \rho(y) \neq \perp, \quad A(y, C) = 0 \quad \text{if } \rho(y) = \perp, \quad (1)$$

so $A = 1$ requires the training gate and every requested family check to pass. At evaluation, all-satisfied rate is the fraction of all test items whose outputs pass the parse gate and satisfy every requested musical constraint, i.e. the mean of A .

Table 1: The eight constraint families, each decided by a deterministic `music21` routine on the parsed score, spanning global, bar, note, transition and cross-section properties. The third column is the default sampler’s domain; the verifier domain Θ_f is wider, which the Edge split exploits. Table 18 adds each prompt sentence.

Family	Checked over	Default sampler values
key	global	13 keys: 8 major, 5 minor
meter	every bar	4/4, 3/4, 6/8, 2/4; each bar must sum to the signature
length	global	8, 12 or 16 bars, no pickup
range	every note	4 inclusive pitch spans, e.g. G3–A5
final	last note	a pitch class, any octave (a scale degree 1/3/5 when a key is present)
rhythm	every note	4 admissible duration sets; no dotted values, no rests
leap	adjacent notes	largest melodic interval: 7, 9 or 12 semitones
structure	4-bar units	AABA, AABB or ABAB over 16 bars; equal letters identical, different letters distinct

3.2 CONSTRAINT FAMILIES AND VERIFIERS

Verifying properties at multiple musical scales. Table 1 lists the eight constraint families. Seven constrain attributes other than multi-section form: key, meter, bar count, pitch range, final pitch class, admissible note durations, and largest melodic interval. The eighth, **structure**, is a cross-section form constraint: a sixteen-bar tune in a stated form over four-bar units (AABA, AABB, ABAB), units sharing a letter identical note-for-note and units with different letters differing. A unit u_a ($a \in \{1, \dots, 4\}$) is its sequence of $M = 4$ bar signatures (pitch and duration per event); a tune without exactly $4M$ bars scores 0 on every structure term. Writing E for the index pairs required equal, D for those required different (minimal sets per form in Appendix C.1) and $\mathbb{I}[\cdot]$ for an indicator,

$$v_{\text{structure}} = \mathbb{I}\left[\bigwedge_{(a,b) \in E} u_a = u_b\right] \cdot \mathbb{I}\left[\bigwedge_{(a,b) \in D} u_a \neq u_b\right]. \quad (2)$$

We write v_{rep} and v_{contrast} for the two factors, the *repetition* and *contrast* halves; their diagnostic roles are examined in §5.5. Their conjunction is zero whenever either fails, however close the output is. Writing $u_a[m]$ for the m -th bar signature of unit u_a , the graded counterparts are

$$\bar{e} = \frac{1}{|E|M} \sum_{(a,b) \in E} \sum_{m=1}^M \mathbb{I}[u_a[m] = u_b[m]], \quad \bar{d} = \frac{1}{|D|M} \sum_{(a,b) \in D} \sum_{m=1}^M \mathbb{I}[u_a[m] \neq u_b[m]], \quad (3)$$

In Equation 3, $\bar{e}$ counts how much of the required repetition is already in place rather than whether all of it is. Equation 4 uses $\bar{e}$; Appendix E.2 reports what happens when $\bar{d}$ is used symmetrically alongside it.

Calibrating the key criterion. The key verifier is the one hybrid among the eight, because a literal check fails in both directions: the K: header is gameable, and key-profile analysis (Krumhansl, 2001; Temperley, 1999) (Aarden–Essen profiles) agrees with the corpus header on only 0.750 of 400 corpus tunes. We accept a tune iff the header matches *and* the analysed key is the requested one or its relative (Appendix C.3).

3.3 REWARD

Grading per-family credit behind a hard validation gate. For a completion y on problem C with $k = |C|$,

$$R(y, C) = \begin{cases} 0, & \rho(y) = \perp, \\ \lambda \cdot \frac{1}{k} \sum_{(f, \theta) \in C} s_f(\rho(y), \theta) + (1 - \lambda) A(y, C), & \text{otherwise,} \end{cases} \quad (4)$$

with $\lambda = 0.7$, which leaves the binary term 0.3 of the reward even at $k = 1$ (Appendix D.9 sweeps it). Here $s_f : \mathcal{S} \times \Theta_f \rightarrow [0, 1]$ equals v_f for the seven non-structure families; for **structure** it is graded, $s_{\text{structure}} = 0.7 \bar{e} + 0.3 v_{\text{contrast}}$, where $\bar{e}$ is the fraction of matching bar positions across pairs in E . Grading repetition and leaving contrast binary is a design choice: exact bar matches give one

notion of partial repetition, whereas counting differing bars need not reflect perceptually meaningful contrast (symmetric variant: Appendix E.2). The parse gate is hard: unparseable output earns zero. Equation 4 affects training only; the main comparison and reward-design ablation use the same binary success criterion. Partial training credit is never counted as all-satisfied success.

Partial credit matters because of how GRPO estimates advantages. For a prompt x_C , GRPO samples a group of G rollouts and centres their rewards within that group,

$$\hat{A}_i = \frac{R(y_i, C) - \mu_C}{\sigma_C + \epsilon}, \quad \mu_C = \frac{1}{G} \sum_{j=1}^G R(y_j, C), \quad \sigma_C^2 = \frac{1}{G-1} \sum_{j=1}^G (R(y_j, C) - \mu_C)^2, \quad (5)$$

with $\epsilon = 10^{-4}$ a small constant. A prompt contributes a reward-driven update only when rewards *differ* inside its own group; their level does not enter, and a group whose rollouts all score alike contributes only the KL term (DAPO (Yu et al., 2026) discards such groups; partial credit can make some of them informative). Binary A separates no partial ordering at all. The λ -weighted average separates rollouts satisfying different *numbers of families*, but with $s_f = v_f$ it still scores form as one conjunction. Holding the contrast verdict fixed, the graded $s_{\text{structure}}$ also separates rollouts that fail the form check but match different fractions of bar positions, which is credit *within* a family. Section 5.5 tests this design against binary A at a matched initialisation and budget: over the first 50 updates the graded reward leaves 0.307 of groups without a reward gradient against 0.550 under binary A , and the graded arm ends ahead over the benchmark as a whole. Appendix D.14 reports the full traces and Appendix E.2 the form decomposition.

Rejecting a malformed output rather than paying for its recoverable fraction. Scoring that fraction (cf. Skalse et al., 2022) would let a fragment satisfying key and range, but forming no tune, collect $\lambda \cdot 2/k$. Training and main evaluation share the gate; Appendix C.2 audits additional fixed-format requirements and re-scores the matched checkpoints of Table 2 under them, all of which keep every success on all eight suites. The family verifiers and score parser encode music-specific assumptions; the reward aggregation and optimiser are generic.

3.4 TRAINING

We compare two initialisations: *Direct* starts from Qwen3-4B-Instruct; *SFT init.* first applies the music-domain SFT below. Both then use the same reward, prompt pool and GRPO settings. The initial policy also serves as the KL reference, so the route comparison includes both initialisation and reference-policy changes.

Optimising with the verifier as the only source of task supervision. Both routes use GRPO (Shao et al., 2024). For each prompt x_C in the reinforcement pool, the behaviour policy $\pi_{\phi_{\text{old}}}$ samples a group $\{y_1, \dots, y_G\}$, every rollout is scored by Equation 4, and the group-relative advantage of Equation 5 is assigned to every token of its rollout. Writing $r_{i,t}(\phi)$ for the token-level likelihood ratio $\pi_{\phi}(y_{i,t} \mid x_C, y_{i,<t}) / \pi_{\phi_{\text{old}}}(y_{i,t} \mid x_C, y_{i,<t})$, the objective is the PPO clipped surrogate (Schulman et al., 2017) with a reference-model penalty,

$$\mathcal{J}(\phi) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left\{ \min(r_{i,t} \hat{A}_i, \text{clip}(r_{i,t}, 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}}) \hat{A}_i) - \beta \hat{K}_{i,t} \right\} \right], \quad (6)$$

where the expectation is over prompts $C \sim \mathcal{D}_{\text{RL}}$ and their sampled groups, $|y_i|$ counts completion tokens, $\varepsilon_{\text{clip}} = 0.2$ is the clipping range, $\hat{K}_{i,t} = e^d - d - 1$ with $d = \log \pi_{\text{ref}}(y_{i,t} \mid \cdot) - \log \pi_{\phi}(y_{i,t} \mid \cdot)$ is the per-token KL estimator of Shao et al. (2024), $\pi_{\phi_{\text{old}}}$ is a detached copy of π_{ϕ} (one update per rollout batch), and π_{ref} is the route’s initial checkpoint (Appendices A and B.2). No value network is trained; the group mean μ_C is the baseline, which leaves the verifier as the only source of task-specific supervision in the loop. We run $G = 8$, $\beta = 0.04$, learning rate 1×10^{-6} , 300 optimiser steps on 6,000 synthetic prompts, with Qwen3-4B-Instruct (Yang et al., 2025) trained full-parameter and rollouts served by vLLM on a dedicated device (§B.1). The prompt pool upsamples structure-containing problems to 33% (against 21% unforced), which changes the prompt distribution as well as exposure to form; its control is in §5.5 and the full configuration in Table 5.

Warm-starting on targets that the verifiers label themselves. The optional SFT uses 12,890 instruction–tune pairs self-labelled from the IrishMAN corpus (Wu et al., 2023): for each corpus tune the verifiers read candidate constraint parameters off the parsed score, sample $k \sim \mathcal{U}\{1, 2, 3\}$ families, and admit the pair only if $A(y, C(y)) = 1$. Held-out family pairs (§4) are excluded from both training pools. Full construction details are in Appendices B.3–B.5.

4 MUSICCONSTRAINTBENCH

MUSICCONSTRAINTBENCH comprises five complementary tests: *constraint density* (Single, Mixed, High- k), *compositional generalisation* (Seen, Unseen, Dense), *parameter extrapolation* (Edge), *multi-section control* (Structure), and *linguistic variation* (paraphrases). Single ($n = 200$) contains one constraint, High- k ($n = 160$) four, and Mixed ($n = 300$) spans $k \in \{1, 2, 3, 4\}$ and serves as the per-family diagnostic. Seen ($n = 200$) contains two or three families that co-occur in training, whereas Unseen ($n = 200$) requires one of four held-out pairs; Structure ($n = 120$) always includes the cross-section form constraint. Dense and Edge contain 500 items each (§5.4), giving 2,180 items in total. The six original suites were consulted during reward development, whereas Dense and Edge were added afterwards. The held-out pairs are (key, rhythm), (meter, leap), (final, structure), and (length, range); none occurs together in either training stage (Lake & Baroni, 2018; Kim & Linzen, 2020). They form a perfect matching over the eight families, so each family is held out once and any held-out-pair-free set has at most $k = 4$ (Appendix C.4). Finally, re-rendering all Single and Seen items with alternative wording yields a 400-item paraphrase suite with unchanged constraint values (Appendices C.8, C, B.6, and C.6).

5 EXPERIMENTS

In this section, we address five questions: **Q1**, How well existing models follow combinations of musical instructions; **Q2**, Whether verifier-driven RL can train a general-purpose model beyond them without requiring music-domain SFT; **Q3**, Whether the gains generalise to excluded combinations, extrapolated parameters, and reworded prompts; **Q4**, Why the reward design works; and **Q5**, What the resulting compliance score captures. Sections 5.2–5.6 answer these questions in order. The principal comparisons use the matched runs of Table 2, while additional ablations and diagnostics use a separate diagnostic study (Appendices D–E).

5.1 EXPERIMENT SETUP

We score saved greedy completions using the all-satisfied criterion of §3; repeated prompts retain their sampler weight, supplementary audits use additional criteria where noted, and the sampling-budget study uses temperature 1.0. Baselines include Qwen3-4B/8B/32B (Yang et al., 2025), Llama-3.1-70B (Grattafiori et al., 2024), MiniMax-M3, and ChatMusician-7B (Yuan et al., 2024), all given the same user prompt under their own chat templates without requested reasoning; the music-domain SFT checkpoint serves as a supervised baseline (Appendix B.4). The three *matched runs*, Direct binary ($R = A$), Direct MUSICRLVR, and SFT-initialised MUSICRLVR, share seed 43, 300 updates, and 48 completions per update. The two Direct arms start from Qwen3-4B-Instruct and differ only in reward mixing weight, while SFT initialisation also changes the fixed KL reference. Final checkpoints are evaluated once per suite under greedy decoding, with decoding variation analysed separately (Appendix B.9) and paired McNemar tests (McNemar, 1947) reported in Appendix D.5. Component ablations, sensitivity analyses, paraphrases, and output analyses use a separate SFT-initialised reference, termed the *diagnostic study*, unless stated otherwise.

5.2 A1: EXISTING MODELS STRUGGLE WITH MULTI-CONSTRAINT MUSIC INSTRUCTIONS

For the strongest zero-shot models, parsing is largely solved; joint compliance is not. As shown in Table 2, Llama-3.1-70B writes a parseable score for 0.970 of Mixed items yet satisfies every requested constraint on only 0.380, with Qwen3-32B and MiniMax-M3 alongside it at 0.363 and 0.367: the three strongest zero-shot models sit within 0.017 of one another. Nor does capacity order the field: Qwen3-8B reaches 0.123 on Mixed, below the 0.160 of the smaller Qwen3-4B.

Table 2: All-satisfied rate on the eight MUSICCONSTRAINTBENCH suites (greedy decoding); the three RL rows are the matched runs at equal budget. Each column marks its **best** and **runner-up**; ranks are descriptive. High- k and Dense both use $k = 4$ but differ in held-out pairs and family mixture. Counts, parse-gate rates and paired tests: Appendix D.5.

Model	Six original splits						Later splits	
	Mixed	Single	Seen	Unseen	High- k	Structure	Dense	Edge
<i>Zero-shot</i>								
Qwen3-4B	0.160	0.225	0.100	0.040	0.037	0.100	0.008	0.010
Qwen3-8B	0.123	0.210	0.115	0.050	0.006	0.000	0.020	0.032
Qwen3-32B	0.363	0.535	0.250	0.190	0.094	0.267	0.074	0.112
Llama-3.1-70B	0.380	0.695	0.300	0.195	0.056	0.167	0.040	0.058
MiniMax-M3	0.367	0.580	0.315	0.190	0.100	0.008	0.088	0.160
ChatMusician-7B	0.160	0.290	0.130	0.090	0.006	0.000	0.020	0.046
<i>Trained (4B)</i>								
SFT (4B)	0.477	0.745	0.415	0.440	0.231	0.600	0.290	0.076
Direct GRPO ($R = A$)	0.797	0.960	0.745	0.565	0.469	0.908	0.322	0.300
MusicRLVR (SFT init.)	0.577	0.880	0.585	0.530	0.350	0.383	0.366	0.134
MusicRLVR (Direct)	0.807	0.985	0.815	0.595	0.575	0.967	0.362	0.310

Where compliance is lost is visible by constraint count (Figure 1), each k slice carrying its own prompts and family mixture. Llama-3.1-70B satisfies 0.720 of single-constraint Mixed items, as many as the music-domain SFT checkpoint, but only 0.050 of four-constraint ones (Appendix D.11): a $14.4\times$ fall, where Direct MusicRLVR falls $2.0\times$. Strength on one instruction at a time therefore says little about honouring several together, which is the regime an explicit specification lives in. Music specialisation does not substitute for it either, ChatMusician-7B reaching merely 0.160 on Mixed under the shared prompting protocol (Appendix D.2).

5.3 A2: DIRECT VERIFIER-DRIVEN RL TEACHES CONSTRAINT FOLLOWING WITHOUT MUSIC-DOMAIN SFT

Direct MusicRLVR raises the Mixed all-satisfied rate from 0.160 to 0.807 and surpasses every evaluated zero-shot model on all eight suites — 0.985 on Single and 0.967 on Structure (both in-distribution, Appendix B.7), 0.575 on High- k against a best zero-shot of merely 0.100 (Table 2) — with every completion clearing the parse gate on the six original suites. Music-domain SFT is not a prerequisite: under the same reward, prompt pool and budget, the SFT-initialised route reaches 0.577 on Mixed against 0.807 for Direct and trails on seven of eight suites, Dense excepted at 183/500 against 181/500; pooled over the 2,180 items they disagree on 767, 554 in Direct’s favour ($p < 10^{-4}$), six of the eight per-suite advantages surviving Holm correction (Appendix D.5). From these results we observe that (1) music-domain SFT is not required for the observed gains; (2) the supervised initialisation is the arm that collapses on form, clearing the repetition half of the form verifier on 119/120 Structure items but the contrast half on only 50, and emitting an all-identical tune on 69 (Figure 4(e)); and (3) the deficit manifests during training, where over the last 50 updates that arm satisfies everything on 0.751 of its rollouts against 0.890 for Direct. Initialisation also fixes the KL reference (Figure 4(c,f)). Figure 3 illustrates two verifier-successful compositions meeting the same four constraints with different repeating patterns (Appendices F and D.16).

5.4 A3: THE GAIN GENERALISES BEYOND TRAINING COMBINATIONS AND PARAMETER RANGES

Compositional generalisation. Every Unseen item requires one of the four family pairs excluded from both training pools; Direct reaches 0.595 there against 0.815 on Seen, where the best zero-shot rates are 0.195 and 0.315. Both four-constraint suites hold up too, 0.575 on High- k and 0.362 on Dense, which also requires a held-out pair and differs in family mixture (Appendix B.7).

Parameter extrapolation. Edge moves meters to $5/4$, $2/2$, $9/8$, $12/8$ and lengths to 4, 24, 32 bars and extends leap, range and rhythm (Table 12), sharing the sampler and verifiers with the standard suites. Direct scores 0.310 against the strongest zero-shot 0.160 and the SFT checkpoint’s 0.076. Appendix D.3 separately analyses pair-free and held-out-pair subsets for the diagnostic SFT-initialised checkpoint.

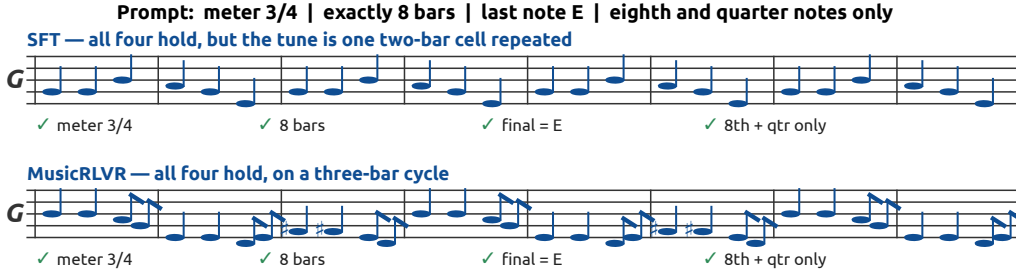


Figure 3: **One high- k item from the diagnostic study** (the checkpoints of Appendix D.6, not the matched rows of Table 2). Both outputs satisfy the four constraints, but SFT repeats one two-bar cell while SFT-initialised MusicRLVR uses a three-bar cycle. Verifier success permits different compositions; the example makes no perceptual-quality comparison.

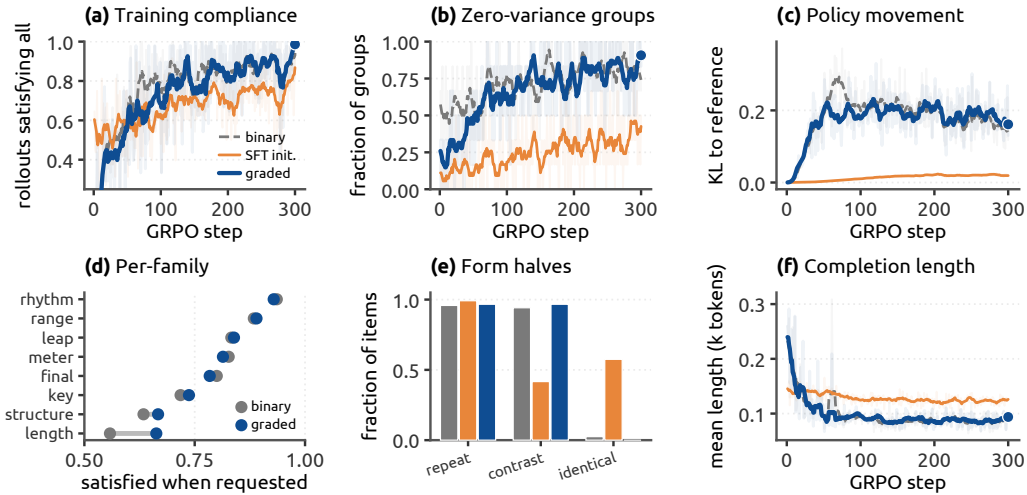


Figure 4: **Matched runs in Table 2.** “Binary” and “graded” are the two Direct arms, differing only in the reward weight λ ; “SFT init.” uses the graded reward from the music-domain SFT checkpoint. (a) All-satisfied rollout rate. (b) Zero-variance groups. (c) KL to each run’s frozen reference. (d) Per-family satisfaction (Direct arms only). (e) Form-verifier components and all-identical rate. (f) Mean completion length.

Linguistic robustness. Rewording each single and seen-combination item while holding its constraint values moves MusicRLVR from 0.735 to 0.688 and the SFT checkpoint from 0.580 to 0.535, losses of 19 and 18 out of 400: the margin carries over intact (diagnostic study; Appendix D.1).

5.5 A4: WHY THE REWARD WORKS: PARTIAL CREDIT BEHIND A VALIDITY GATE

A binary reward leaves many groups without a learning signal: Equation 5 gives no reward gradient when all rollouts score alike, and over the first 50 updates this occurs for 0.550 of binary groups against 0.307 with graded credit (Figure 4(b)). Both arms finish at similar rollout compliance (0.890 versus 0.884), suggesting that graded ordering improves transfer rather than training-set fit. Across the 2,180 benchmark items, graded credit satisfies 1265 against 1188, with 229 versus 152 discordant wins ($p = 9.39 \times 10^{-5}$). Its largest gains occur on High- k (0.469 $\rightarrow$ 0.575) and Seen (0.745 $\rightarrow$ 0.815); it is higher on all eight suites, although the 0.010 margins on Mixed and Edge lie within the re-evaluation spread (Appendix B.9). With each family scored only on the items requesting it, gains concentrate on length (0.558 $\rightarrow$ 0.663, $p < 10^{-4}$) and structure (0.634 $\rightarrow$ 0.667), with the other six moving by at most 0.019 (Figure 4(d); Appendix D.10). In the diagnostic ablations, increasing form exposure from 21% to 33% raises Structure compliance from 0.250 to 0.525, while graded form credit at the same share further raises it to 0.808, with the other five original suites changing by at most 0.025 (Table 3; Appendices D.4 and E.3). The matched runs show the same pattern, with 0 all-identical Structure outputs under graded credit against 3 under binary credit. Further sensitivity and trainer-log analyses are in Appendices D.8–D.15.

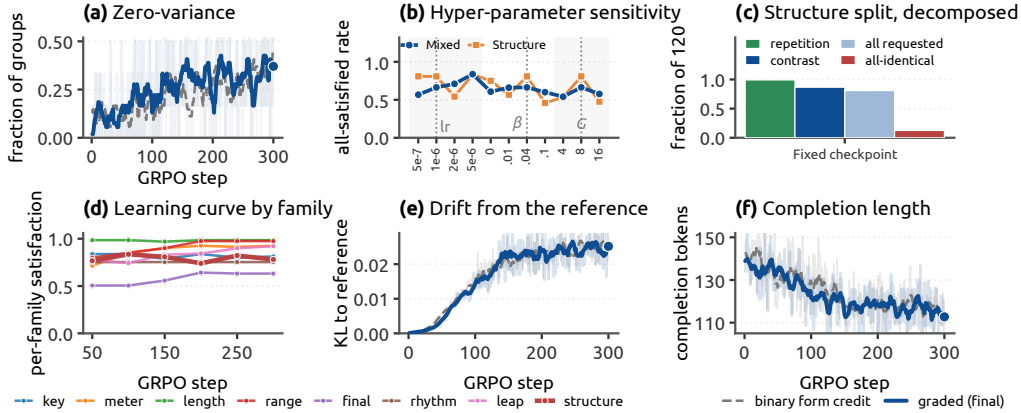


Figure 5: **Reward-design ablations (diagnostic study).** (a) Zero-reward-variance groups. (b) Sensitivity to learning rate, β and group size G ; dotted lines mark the reference settings. (c) The form verifier decomposed. (d) Per-family satisfaction. (e–f) KL and completion length, binary against graded form credit.

Table 3: **Reward-design ablations (diagnostic study).** Each RL row is one 300-step run from the same supervised checkpoint, scored by the binary all-satisfied criterion; the format audit leaves all five unchanged. The two changes separating our final reward from the first attempt each recover about half of the structure gap, and across the first three RL rows the other five splits vary by at most 0.053.

Configuration	Mixed	Single	Seen	Unseen	High- k	Struct.
SFT only, no GRPO	0.477	0.745	0.415	0.440	0.231	0.600
+ GRPO, varying the form-constraint reward and the share of form problems:						
binary form credit, unforced share (21%)	0.610	0.810	0.630	0.635	0.350	0.250
binary form credit, share 33%	0.653	0.810	0.610	0.610	0.331	0.525
graded form credit, share 33% (MusicRLVR)	0.663	0.835	0.635	0.595	0.356	0.808
symmetric form credit, single run (App. E.2)	0.590	0.745	0.590	0.630	0.388	0.200

5.6 A5: WHAT THE VERIFIER CHANGES, AND WHAT IT DOES NOT MEASURE

A compliance score is not a quality score. In the diagnostic study the SFT checkpoint and SFT-initialised RL write tunes of different lengths (39.2 against 28.7 bars on average), yet on the first eight bars of the 297 items where both are long enough their surface statistics move little, the largest shift being pitch-class entropy $1.930 \rightarrow 1.826$ (Table 4a; Appendices D.12, D.13); perceptual quality would require an independent listening study. What the verifier still leaves unsolved is shallow rather than pervasive: Direct misses 58 of 300 Mixed items, of which 48 fall one family short and 10 two or more, with **structure** the most common sole miss (18); none fails the parse gate (Table 32). Form stands near its ceiling on both halves (Figure 4(e)). The verdicts also serve at decoding time: verifier selection among eight samples raises the diagnostic Mixed rate to 0.950 (Table 4b).

Table 4: **Diagnostic output analysis.** (a) Surface statistics (first 8 bars, 297 matched items). (b) Best-of- N ($T=1.0$, $n=8$, binary).

(a)					
Model	Notes	PC ent.	Dist. bars	Dist. 4-gr.	
SFT	42.7	1.930	0.417	0.419	
MusicRLVR (diag.)	40.5	1.826	0.426	0.403	
(b)					
Suite	Model	p@1	p@2	p@4	p@8
Mixed	SFT	0.415	0.584	0.728	0.827
	MusicRLVR	0.676	0.817	0.900	0.950
Struct.	SFT	0.692	0.861	0.938	0.967
	MusicRLVR	0.844	0.953	0.978	0.983

6 CONCLUSION

We frame constraint-following music generation as a *property-verifiable generation* problem and introduce MUSICCONSTRAINTBENCH and MUSICRLVR. Graded property-level rewards behind a hard validity gate raise Qwen3-4B-Instruct from 0.160 to 0.807 on Mixed without music-domain SFT, with gains extending to unseen property combinations and out-of-range parameters. These results show that deterministic property checks can provide effective task-specific supervision for open-ended structured generation without requiring a target composition.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks with required disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for polishing the writing and improving the readability of the manuscript. We have reviewed all AI-assisted work: all AI-assisted edits were checked against the underlying results by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The study involves no human subjects, no personal data and no user-facing deployment. Supervised targets derive from the publicly available IrishMAN corpus of traditional Irish music (Wu et al., 2023); the reinforcement prompts are produced by deterministic samplers rather than drawn from a corpus. The intended uses are settings that need a score to meet a stated specification, such as music education, game audio and accessibility tools. We report compliance with programmatic checks and make no claim about musical quality; reading a compliance score as a quality score would overstate what the verifiers measure, which is the misuse we consider most likely. Appendix G gives the longer discussion.

REPRODUCIBILITY STATEMENT

Quantitative tables, macros and figures are regenerated from saved evaluation files and training logs by the scripts of Appendix B.8; the principal training settings are in Appendix B, the verifiers in Appendix C.1, and Appendix B.10 lists the artefacts we will release.

REFERENCES

- Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.
- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.
- Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, et al. Musicrl: Aligning music generation to human preferences. *arXiv preprint arXiv:2402.04229*, 2024.
- Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in neural information processing systems*, 36:47704–47720, 2023.
- Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

- Xuzheng He, Nan Nan, Zhilin Wang, Ziyue Kang, Zhuoru Mo, Ao Li, Yu Pan, Xiaobing Li, Feng Yu, and Xiaohong Guan. Symphonygen: 3d hierarchical orchestral generation with controllable harmony skeleton. *arXiv preprint arXiv:2604.25498*, 2026.
- Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.
- Yujia Huang, Adishree Ghatare, Yuanzhe Liu, Ziniu Hu, Qinsheng Zhang, Chandramouli S Sastry, Siddharth Gururani, Sageev Oore, and Yisong Yue. Symbolic music generation with non-differentiable rule guided diffusion. *arXiv preprint arXiv:2402.14285*, 2024.
- Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pp. 1645–1654. PMLR, 2017.
- Nicolas Jonason, Luca Casini, and Bob LT Sturm. Smart: Tuning a symbolic music generation system with an audio domain aesthetic reward. *arXiv preprint arXiv:2504.16839*, 2025.
- Maximos Kaliakatos-Papakostas, Konstantinos Soiledis, Theodoros Tsamis, Dimos Makris, Vassilis Katsouros, and Emiliios Cambouropoulos. Incorporating structure and chord constraints in symbolic transformer-based melodic harmonization. *arXiv preprint arXiv:2512.07627*, 2025.
- Najoung Kim and Tal Linzen. Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pp. 9087–9105, 2020.
- Carol L Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, 2001.
- Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pp. 2873–2882. PMLR, 2018.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pp. 4288–4299, 2021.
- Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhui Chen. General-reasoner: Advancing llm reasoning across all domains. *Advances in Neural Information Processing Systems*, 38:56596–56618, 2026.
- Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.
- Xingwei Qu, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xeron Du, et al. Mupt: A generative symbolic music pretrained transformer. In *International Conference on Learning Representations*, volume 2025, pp. 46753–46779, 2025.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Joar Skalse, Nikolaus Howe, Dmitrii Krashenninikov, and David Krueger. Defining and characterizing reward gaming. *Advances in neural information processing systems*, 35:9460–9471, 2022.
- Bob L Sturm, Joao Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. Music transcription modelling and composition using deep learning. *arXiv preprint arXiv:1604.08723*, 2016.
- David Temperley. What’s key for key? the krumhansl-schmuckler key-finding algorithm reconsidered. *Music Perception*, 17(1):65–100, 1999.
- John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *arXiv preprint arXiv:2306.08620*, 2023.
- Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. Figaro: Controllable music generation using learned and expert features. In *The Eleventh International Conference on Learning Representations*, 2023.
- Yashan Wang, Shangda Wu, Jianhuai Hu, Xingjian Du, Yueqi Peng, Yongxin Huang, Shuai Fan, Xiaobing Li, Feng Yu, and Maosong Sun. Notagen: Advancing musicality in symbolic music generation with large language model training paradigms. *arXiv preprint arXiv:2502.18008*, 2025a.
- Zhilin Wang, Zhe Yang, Yun Luo, Yafu Li, Xiaoye Qu, Ziqian Qiao, Haoran Zhang, Runzhe Zhan, Derek F Wong, Jizhe Zhou, et al. Towards an ai musician: Synthesizing sheet music problems for musical reasoning. *arXiv preprint arXiv:2509.04059*, 2025b.
- Shangda Wu, Xiaobing Li, Feng Yu, and Maosong Sun. Tunesformer: Forming irish tunes with control codes by bar patching. *arXiv preprint arXiv:2301.02884*, 2023.
- Shangda Wu, Yashan Wang, Ruibin Yuan, Guo Zhancheng, Xu Tan, Ge Zhang, Monan Zhou, Jing Chen, Xuefeng Mu, Yuejie Gao, et al. Clamp 2: Multimodal music information retrieval across 101 languages using large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 435–451, 2025.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.
- Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, et al. Chatmusician: Understanding and generating music intrinsically with llm. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6252–6271, 2024.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.
- Alon Ziv, Sanyuan Chen, Andros Tjandra, Yossi Adi, Wei-Ning Hsu, and Bowen Shi. Mr-flowdpo: Multi-reward direct preference optimization for flow-matching text-to-music generation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1821–1825. IEEE, 2026.

CONTENTS OF APPENDIX

A	Algorithm	15
B	Experimental Settings	16
B.1	Infrastructure	16
B.2	Training Hyperparameters	16
B.3	SFT Data Construction	17
B.4	SFT Training Examples	17
B.5	Prompt Template	18
B.6	Dataset Statistics	19
B.7	Prompt Coverage, Training Overlap, and Split Composition	19
B.8	Reproducibility	21
B.9	Evaluation Nondeterminism	21
B.10	Data and Code Release	22
C	Benchmark and Verifier Details	22
C.1	Verifier Implementation Details	22
C.2	Stricter Fixed-Instruction Gate Audit	23
C.3	Key Verifier Calibration	25
C.4	Held-out Pair Rationale	25
C.5	ABC Notation Primer	26
C.6	ABC Notation: A Worked Example	26
C.7	Constraint Rendering Examples	27
C.8	Paraphrase Examples	28
D	Additional Experiments	28
D.1	Paraphrase and Training Dynamics	29
D.2	Rejection Sampling and a Music-Specialised Baseline	29
D.3	Edge Subsets and a Stricter Key Criterion	30
D.4	Significance Tests	31
D.5	Matched Comparison	31
D.6	Diagnostic Reference Results	33
D.7	Fixed-Checkpoint Evaluation	34
D.8	Hyper-parameter Sweeps	34
D.9	Reward Mixing Weight	35
D.10	Per-Family Satisfaction on the Original Suites	36
D.11	Performance by Number of Constraints (k)	37
D.12	Output Length Statistics	37
D.13	Length-Controlled Output Statistics	37
D.14	Zero-Variance Groups During Training	39
D.15	Additional Training Curves	39
D.16	Additional Diagnostics	40

E Error and Structure Analysis	40
E.1 Which Constraint Binds, by Split	40
E.2 Structure Reward Variants	41
E.3 Structure Results by Form Type	42
F Qualitative Examples	43
G Broader Impact	47

A ALGORITHM

Algorithm 1 runs verifier-driven GRPO directly or with an optional supervised warm start built from verified corpus-derived labels ($\theta_f(y) := p_f(\rho(y))$, §3.4). Unless marked Direct, the diagnostics below describe the SFT-initialised route, retaining a common initialisation for the reward-design comparisons.

Algorithm 1: MusicRLVR: verifier-only GRPO with optional self-labelled SFT

Input: initial policy π_{ϕ_0} ; optional SFT flag u and corpus tunes $\mathcal{Y}$; families $\mathcal{F}$ with verifiers $\{v_f\}$ and parser ρ ; held-out pairs H ; held-out retries N ; reinforcement pool $\mathcal{D}_{\text{RL}}$ (no prompt contains a pair in H ; structure share 33%); reward weight λ ; group size G ; β , $\varepsilon_{\text{clip}}$, ϵ ; steps T ; learning-rate schedule η_s

Output: trained policy π_{ϕ}

```

 $\pi_{\text{ref}} \leftarrow \pi_{\phi_0}$ 
if  $u$  is enabled then
  // Optional self-labelled supervised fine-tuning (§3.4)
   $\mathcal{D} \leftarrow \emptyset$ 
  foreach  $y \in \mathcal{Y}$  do
     $C(y) \leftarrow \text{NULL}$ ; draw  $k \sim \mathcal{U}\{1, 2, 3\}$ 
    for  $\text{attempt} = 1, \dots, N$  do
      sample  $F \subseteq \mathcal{F}_y$  (families measurable on  $y$ ),  $|F| = \min(k, |\mathcal{F}_y|)$ 
       $C \leftarrow \{(f, \theta_f(y)) : f \in F\}$  with  $\theta_f(y) = p_f(\rho(y))$  read off  $\rho(y)$ 
      if  $C$  contains no pair in  $H$  then  $C(y) \leftarrow C$ ; break
    end
    if  $C(y) = \text{NULL}$  then
       $C(y) \leftarrow$  one constraint drawn from  $y$ 's properties // singleton: trivially held-out-free
    end
    if  $A(y, C(y)) = 1$  then
       $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{C(y)}, y)\}$  // else this draw of  $y$  is discarded
    end
    // with prob. 0.2: a structure-only pair from  $y$ 's first two 4-bar units, re-verified the same way
  end
   $\pi_{\text{ref}} \leftarrow$  fine-tune  $\pi_{\phi_0}$  on  $\mathcal{D}$ 
end
 $\pi_{\phi} \leftarrow \pi_{\text{ref}}$ 
// Verifier-driven GRPO (§3.4)
for  $\text{step } s = 1, \dots, T$  do
   $\phi_{\text{old}} \leftarrow \phi$  (detached)
  sample a batch of prompts  $x_C$ ,  $C \sim \mathcal{D}_{\text{RL}}$ 
  foreach  $x_C$  in the batch do
    sample a group  $\{y_1, \dots, y_G\} \sim \pi_{\phi_{\text{old}}}(\cdot | x_C)$ 
    score every  $y_i$  by  $R(y_i, C)$  (Equation 4)
     $\hat{A}_i \leftarrow (R(y_i, C) - \mu_C) / (\sigma_C + \epsilon)$  for every  $i$  (Equation 5) //  $\hat{A}_i = 0$  for every  $i$ 
    if  $\sigma_C = 0$ 
  end
  accumulate  $\nabla_{\phi}(-\mathcal{J}(\phi))$  over the batch, clip its norm to 1.0, AdamW step at  $\eta_s$ 
  // Equation 6
end

```

B EXPERIMENTAL SETTINGS

B.1 INFRASTRUCTURE

Training uses ms-swift’s GRPO implementation with a custom reward plugin that shells out to the evaluation verifiers, so training and benchmarking share one `music21` build. On these accelerators, co-locating the vLLM rollout engine with the training state is infeasible for full-parameter 4B training: the sleeping engine’s memory pool still counts against the training process and the backward pass fails at a fixed allocation point regardless of batch size. Serving rollouts from a dedicated device solves it. Two further settings are required on this stack: disabling the NZ tensor layout (it corrupts weight synchronisation during RL) and capping the rollout engine’s context length, which otherwise reserves key-value cache for the model’s full 262,144-token window.

B.2 TRAINING HYPERPARAMETERS

Table 5 preserves the optional SFT and seed-42 diagnostic GRPO settings. The matched runs use two policy devices with accumulation 12 and one rollout device (memory fraction 0.92), keeping 48 completions per update; see Appendix D.5. The saved trainer configurations carry the remaining optimiser defaults.

Table 5: Optional SFT and seed-42 diagnostic GRPO hyperparameters. TP: tensor-parallel degree; mem. frac.: the rollout engine’s memory fraction as recorded in the saved trainer configuration.

	Optional SFT	GRPO
Base model	Qwen3-4B-Instruct	
Parameters trained	Full	Full
Training data	12,890 pairs	6,000 prompts
Optimiser	AdamW	AdamW
Learning rate	1×10^{-5}	1×10^{-6}
LR schedule	Cosine	Cosine
Warmup (fraction of steps)	3%	2%
Batch size (per device $\times$ devices)	4×3 pairs	2×3 completions
Gradient accumulation	4	8
Effective batch per optimiser step	48 pairs	48 completions (6 groups)
Epochs / steps	2 epochs (538 steps)	300 steps
Max sequence length	1536	1536 (completion ≤ 768)
Group size (G)	—	8
KL coefficient (β)	—	0.04
Clipping range ($\varepsilon_{\text{clip}}$)	—	0.2
Rollout temperature	—	1.0
λ (Equation 4)	—	0.7
Structure share in data	16.5%	33.1% (upsampled from 21%)
Precision	bf16	bf16
Devices	3	3 train + 1 rollout
Rollout engine	—	vLLM (TP = 1, mem. frac. 0.9)
Framework	ms-swift 4.2.3	ms-swift 4.2.3
Wall time	~45 min	~3 hours

Figure 6 shows the supervised stage; the reported run uses the fixed two-epoch budget, without testing whether a longer supervised run would improve downstream compliance. Two diagnostics from the GRPO logs are worth recording alongside it. The clipped fraction of Equation 6 is identically zero at all 300 steps, as expected when one optimiser update follows each rollout batch, so the clipping range never binds. In the default diagnostic GRPO run no completion reaches the 768-token cap: the longest sampled completion over training is 492 tokens, and the trainer’s truncation counter stays at zero.

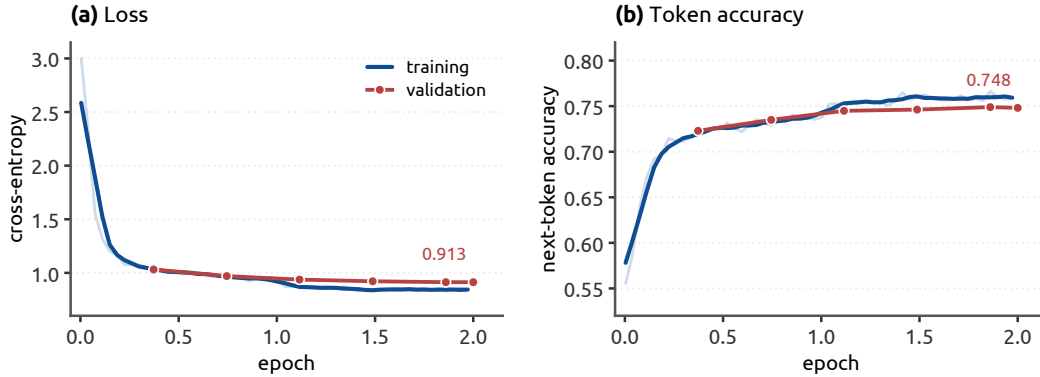


Figure 6: **The supervised stage.** Cross-entropy (a) and next-token accuracy (b) over the two SFT epochs, for the 12,890 training pairs and the 125-pair validation split. Validation loss decreases to 0.913 and validation accuracy plateaus at 0.748.

B.3 SFT DATA CONSTRUCTION

The supervised training data is built from the IrishMAN corpus (Wu et al., 2023), a collection of ~45,000 traditional Irish tunes in ABC notation. The construction pipeline has four steps:

1. **Filter and parse.** Tunes whose body contains the builder’s list of unsupported tokens, or that have fewer than 4 or more than 32 written bars, are discarded; the survivors are parsed with `music21` and kept only if the parsed bar count agrees with the written one.
2. **Property extraction.** The same verifiers used for evaluation extract the true properties of each tune: its key (hybrid policy), meter, bar count, pitch range, final note, the set of distinct note durations, and the largest melodic interval.
3. **Instruction rendering.** A random subset of the extracted properties is selected (with k sampled uniformly from $\{1, 2, 3\}$, capped by the number of measurable properties) and rendered into a constraint instruction using the same template as evaluation. The tune itself is the target output.
4. **Verification gate.** Every (instruction, tune) pair is re-verified: the tune is checked against the rendered constraints using the evaluation verifiers. Any pair that fails verification is discarded. This ensures that no training example teaches the model to produce output that violates its own constraints.

Form examples (structure family) are constructed separately: for a tune with at least eight bars whose first two four-bar units differ, the two units are arranged into a sampled AABA/AABB/ABAB shape (attempted with probability 0.2 per tune; the resulting pair carries the structure constraint only). The resulting 12,890 pairs form the training set, with a further 125 pairs held out for validation. Importantly, the four held-out family pairs (§4) are enforced during construction: any pair whose constraints contain a held-out combination is regenerated with a different family subset.

B.4 SFT TRAINING EXAMPLES

We provide three representative (instruction, output) pairs from the SFT training set to illustrate the data quality and the range of constraint combinations.

SFT example 1: single constraint (meter). **Instruction:** Write a single monophonic tune in ABC notation . . . The tune must satisfy ALL of the following constraints: – the meter must be 6/8, and every bar must contain exactly the right total duration.

Target (from IrishMAN):

```
X:89124
T:Tune
L:1/8
M:6/8
K:G
```

GBd gdB | GBd cBA | GBd gdB | cBA edc | GBd gdB | GBd cBA | cBA dcB | cAF G3 |
 DFA cBA | GAB d2 d | DFA cBA | BAG AGE | DFA cBA | GAB d2 d | cBc dcB | cAF G3 |

Every one of the 16 bars contains exactly six eighth-note units, and the tune carries no repeat sign, so it passes the gate and the meter verifier.

SFT example 2: two constraints (length + key). **Instruction:** ... – the tune must be exactly 32 bars long (no pickup bar). – the tune must be in D major.

Target:

X:106003
 T:Tune
 L:1/8
 Q:1/8=180
 M:6/8
 K:D

DFA cdc | AFA BGE | DFA dcd | AFA gfe | DFA cdc | AFA BGE | afd bge | AFD E2 D |
 DFA cdc | AFA BGE | DFA dcd | AFA gfe | DFA cdc | AFA BGE | afd bge | AFD E2 D |
 fef gfg | afd AFD | fef gfg | DFA B2 A | fdf gfg | afd bge | fga Bcd | AFD E2 D |
 fef gfg | afd AFD | fef gfg | DFA B2 A | fdf gfg | afd bge | fga Bcd | AFD E2 D |

The header reads K:D and content analysis returns D major; exactly 32 bars. The constraint parameters were read off this tune’s own properties, and the pair was re-verified before admission.

SFT example 3: structure constraint (structure, AABA). **Instruction:** ... – the tune must be exactly 16 bars with an AABA form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ.

Target (assembled from the source tune’s first two 4-bar units):

X:141183
 T:Tune
 L:1/4
 M:3/4
 K:Bmin
 B2 c | d3 | d2 a | f e d | B2 c | d3 | d2 a | f e d |
 c3 | e3 | f2 B | B3 | B2 c | d3 | d2 a | f e d |

Units 1, 2, and 4 are identical (A sections, bars 1–4, 5–8, 13–16); unit 3 (bars 9–12) differs, satisfying the AABA form. The two units are the source tune’s first eight bars; every assembled pair is verified before entering the dataset.

B.5 PROMPT TEMPLATE

Every benchmark item is rendered into a prompt with the following fixed template. The shared instruction prefix (the first paragraph) is part of the user message of every item; the constraint block is populated from the item’s constraint list. No chain-of-thought, few-shot examples, or reasoning instructions are given.

Write a single monophonic tune in ABC notation.
 Rules: write every bar out explicitly (no repeat signs like |: or :|), one voice only, no chords, no lyrics, no grace notes. Start with an X: header, include T:, M:, L:, K: headers, then the music. Output ONLY the ABC notation and nothing else.
 The tune must satisfy ALL of the following constraints:
 - *<constraint 1 in natural language>*
 - *<constraint 2 in natural language>*
 ⋮

The constraint text for each family is generated by a deterministic renderer that maps parameters to a fixed English sentence. For example, the **key** family with $\theta = (\text{D}, \text{major})$ renders as “*the tune must be in D major*”; the **structure** family with $\theta = (\text{AABA}, 4)$ renders as “*the tune must be exactly 16 bars with an AABA form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ*”. The paraphrase suite (§4) replaces these renderers with alternative phrasings while preserving the constraint values.

B.6 DATASET STATISTICS

Benchmark splits. Table 6 summarises the composition of the six original benchmark splits. The mixed suite spans $k \in \{1, 2, 3, 4\}$ and covers all eight families; the structure split is the only one where every item includes the cross-section form **structure** family. Family frequencies depend on the sampler and exclusion rules (Table 10); the structure split contains its namesake family in every item and one of five companion families in 66 of 120.

Table 6: Composition of MUSICCONSTRAINTBENCH splits: number of items, distribution over k (number of simultaneous constraints), and family coverage, as ranges where shown.

Split	n	k distribution	Families (count)
Mixed	300	1:100, 2:100, 3:60, 4:40	All eight
Single	200	1:200	All eight (16–29 each)
Seen comb.	200	2:116, 3:84	All eight (51–72 each)
Unseen comb.	200	2:88, 3:112	All eight (54–70 each)
High- k	160	4:160	All eight (56–104 each)
Structure	120	1:54, 2:66	structure: 120; 5 others

Training data. The supervised stage uses 12,890 instruction–tune pairs self-labelled from the IrishMAN corpus. The reinforcement stage uses 6,000 synthetic prompts with $k \in \{1, 2, 3\}$ (2,046/3,060/894 respectively). After the reward-design iteration described in §5.5, the share of prompts containing the **structure** family was raised to 33.1% (1,983/6,000) to increase exposure to form constraints; every other family appears in 1,068–1,347 prompts.

B.7 PROMPT COVERAGE, TRAINING OVERLAP, AND SPLIT COMPOSITION

The weighting and fixed- k comparisons below use the diagnostic SFT-initialised reference; the suite definitions and prompt counts apply to both studies.

The suites contain sampled items, not 2,180 distinct instructions: there are 1,623 distinct prompt strings across all eight splits. Table 7 compares each prompt with the user messages in both training datasets. “Set overlap” ignores constraint order and compares family–parameter sets instead. All Single and Structure prompts occur in training, so their scores measure performance on familiar instructions. Unseen, High- k and Dense have no complete prompt or constraint-set overlap. Both training stages also have zero occurrences of the four explicitly held-out pairs. Exclusion concerns requested families; it does not imply that training tunes never possess those properties jointly.

Table 7: Coverage and overlap with either training stage. Overlap columns count sampled test rows, while “unique prompts” counts distinct strings.

Split	Items	Unique prompts	Exact overlap	Set overlap
Mixed	300	222	201	203
Single	200	39	200	200
Seen	200	177	147	151
Unseen	200	159	0	0
High- k	160	156	0	0
Structure	120	50	120	120
Dense	500	490	0	0
Edge	500	437	32	39

Equal weighting of distinct prompts. Table 8 first averages repeated rows for each prompt within one evaluation, then averages over distinct prompts. The RL column uses the same fixed

checkpoint and evaluation files as Table 33. The advantage over SFT persists on all eight splits under this weighting. On Structure, whose 120 items repeat 50 prompts, the rates are 0.637 and 0.522, respectively. Item-paired tests in the main analysis retain the original sampler weights, so they measure the sampled suites rather than 2,180 distinct instructions. Table 9 applies the same weighting to the three matched checkpoints of Table 2, read from the same saved per-item files.

Table 8: Original item weighting and equal weighting of unique prompts, using main-table criterion and the SFT-initialised checkpoint.

Split	SFT, items	SFT, unique	RL, items	RL, unique
Mixed	0.477	0.402	0.663	0.615
Single	0.745	0.705	0.835	0.820
Seen	0.415	0.407	0.635	0.618
Unseen	0.440	0.396	0.595	0.547
High- k	0.231	0.237	0.356	0.359
Structure	0.600	0.522	0.808	0.637
Dense	0.290	0.290	0.370	0.371
Edge	0.076	0.078	0.162	0.164

Table 9: The same weighting applied to the three matched checkpoints of Table 2: “items” repeats the sampler weights of Table 2, “unique” averages within each distinct prompt first. The splits whose prompts repeat most, Single (39 distinct prompts) and Structure (50), move most; the ordering between the three checkpoints is unchanged on every split but Edge, where the two Direct arms swap by 0.008.

Split	Direct GRPO ($R = A$)		MusicRLVR (SFT init.)		MusicRLVR (Direct)	
	items	unique	items	unique	items	unique
Mixed	0.797	0.741	0.577	0.527	0.807	0.760
Single	0.960	0.949	0.880	0.875	0.985	0.981
Seen	0.745	0.729	0.585	0.567	0.815	0.814
Unseen	0.565	0.509	0.530	0.479	0.595	0.547
High- k	0.469	0.458	0.350	0.353	0.575	0.571
Structure	0.908	0.860	0.383	0.457	0.967	0.940
Dense	0.322	0.322	0.366	0.363	0.362	0.358
Edge	0.300	0.291	0.134	0.135	0.310	0.283

Fixed k does not fix family composition. Table 10 gives family frequencies in High- k , Dense and Edge. The first two share $k = 4$ but are independently sampled and have different family mixtures. Edge additionally mixes pair-free and held-out items (171/500 contain a held-out pair). Appendix D.3 scores the pair-free and held-out-pair halves of Edge separately. Table 11 bootstraps the Dense-minus-High- k difference for each model, separating item-sampling variation from this composition difference.

Table 10: Number and percentage of items requesting each family. Percentages sum to more than 100% because items contain multiple families.

Family	High- k ($n = 160$)	Dense ($n = 500$)	Edge ($n = 500$)
key	76 (47.5%)	288 (57.6%)	168 (33.6%)
meter	82 (51.2%)	282 (56.4%)	160 (32.0%)
length	60 (37.5%)	209 (41.8%)	127 (25.4%)
range	100 (62.5%)	236 (47.2%)	164 (32.8%)
final	104 (65.0%)	234 (46.8%)	180 (36.0%)
rhythm	84 (52.5%)	279 (55.8%)	152 (30.4%)
leap	78 (48.8%)	270 (54.0%)	157 (31.4%)
structure	56 (35.0%)	202 (40.4%)	148 (29.6%)

All three intervals include zero. Relative changes such as -6% and -29% are descriptive point estimates, particularly sensitive to the small Llama-70B success counts. Table 10 gives the family composition the two splits differ in.

Table 11: Dense minus High- k : absolute all-satisfied rate differences with percentile intervals from 20,000 within-split item bootstrap replicates (seed 20260909). They quantify item-sampling variation for these fixed checkpoints, not training or serving variation, and do not adjust for the different family compositions.

Model	High- k	Dense	Δ	95% interval
SFT	0.231	0.290	+0.059	$[-0.019, +0.132]$
MusicRLVR (re-evaluated)	0.394	0.370	-0.024	$[-0.111, +0.062]$
Llama-70B	0.056	0.040	-0.016	$[-0.057, +0.021]$

B.8 REPRODUCIBILITY

`make_assets.py` regenerates the original quantitative tables and numerical macros; `make_audit_assets.py` adds the overlap, sensitivity and gate-audit tables; `src/rescore_subsets.py` produces the Edge-subset and stricter-key tables. The historical reporting pass, `make_checkpoint_assets.py -sync`, computes reference-table, by- k , Edge-subset and fixed- k comparisons from saved verdicts, checks ablation equivalence under the format audit, and supplies the fixed-checkpoint decompositions; it also supplies the corresponding historical figure panels. The matched tables, macros and by- k figure are generated by `make_matched_s43_assets.py` after all three runs finish. This script checks saved per-item IDs, success counts and provenance hashes for all 24 evaluations, then renders the two paired comparisons with eight-suite Holm correction. Figure scripts read the same saved reports or training logs. The family-definition and rendering tables are maintained descriptions of the sampler. The six original suites were used during reward development; Dense and Edge were added afterwards, without that exposure. All training and local evaluation ran on Ascend 910B2 accelerators (64 GB HBM each, CANN 8.2.RC1) with vLLM 0.11.0 through `vllm-ascend`; the seed-42 diagnostic GRPO runs used three policy accelerators and one rollout accelerator. The matched study used two policy accelerators and one rollout accelerator. Local models were served at context 4,096 and generated at most 1,200 tokens per item, greedy. The hosted baselines were queried at temperature 0 through the NVIDIA NIM API (`integrate.api.nvidia.com`), MiniMax-M3 as `minimaxai/minimax-m3` in August 2026, with per-model completion budgets of 800 tokens for Llama-3.1-70B, 4,000 for MiniMax-M3 and 2,048 of context for ChatMusician-7B; budgets and timeouts per model are in the release scripts. The release includes per-item completions, original and strict gate decisions, aggregate reports, audit scripts, and training logs. Historical reports remain unchanged; `src/verifiers_legacy.py` reproduces the training and main evaluation metric, while `src/verifiers.py` implements the stricter gate. The reward plugin pins the historical gate unless strict-gate training is explicitly selected. `src/run_verify.py` defaults to this same A criterion (legacy mode).

Table 12: Parameter values of the Edge split against the standard suites. Families are sampled by the same generator; only the value sets differ. `key`, `final` and `structure` take the same values in both, so they are omitted.

Family	Standard suites	Edge
<code>meter</code>	2/4, 3/4, 4/4, 6/8	5/4, 2/2, 9/8, 12/8
<code>length</code>	8, 12, 16 bars	4, 24, 32 bars
<code>leap</code>	$\leq 7, 9, 12$ semitones	$\leq 2, 3, 4$ semitones
<code>rhythm</code>	three- and two-duration sets	{16th, 8th}, {quarter, half}, {quarter}, {half}
<code>range</code>	mixed spans	one-octave spans (C4–C5, D4–D5, E4–E5, G3–G4, A3–A4)

B.9 EVALUATION NONDETERMINISM

Compliance evaluation uses temperature 0, yet vLLM’s batched decoding is not bitwise reproducible. What we measured is the outcome, not the cause: evaluating the diagnostic checkpoint (`outputs/grpo_4b_v2/checkpoint-300`) twice, on 08-22 and 08-24, with identical serving flags (Table 13), none of the 160 high- k completions was byte-identical across the two runs. A plausible mechanism is that the ordering of floating-point reductions depends on batch composition, and that the first generated token (the arbitrary X : reference number) sits on a near-tie between many candidates, so a small numerical difference there can send the whole continuation down a different path. The reported spread is measured directly (Table 13) and does not rest on this account.

Table 13: The same checkpoint evaluated twice under greedy decoding. “Flipped” counts items whose binary all-satisfied outcome differed between the two. Both have parse rate 1.000.

Split	n	First	Second	Flipped	0→1	1→0
Mixed	300	0.663	0.703	52 (17.3%)	32	20
High- k	160	0.356	0.394	32 (20.0%)	19	13
Structure	120	0.808	0.758	26 (21.7%)	10	16

The two evaluations differ by 0.040 on Mixed and 0.050 on Structure. These are observed changes for the same checkpoint under the stated serving configuration; two evaluations do not establish a universal noise bound.

The first evaluation supplies the historical reference and diagnostic values except Dense and Edge, which come from the re-evaluation, and Figure 5(d), which uses a separate full-retention run. The fixed- k comparison uses the re-evaluated High- k score (0.394 → 0.370). Serving one request at a time is a possible way to reduce batch-dependent numerical variation, but determinism must be verified for the specific hardware and serving stack; we do not claim a universal noise bound. Decoding variation enters each arm as item-level discordance that is close to symmetric, which moves the McNemar statistic towards its null rather than away from it; with respect to this source of variation the pooled comparisons of Appendix D.5 are therefore conservative.

B.10 DATA AND CODE RELEASE

We will release the code and synthetic benchmark suites under an MIT licence, with corpus-derived data and model artefacts subject to their upstream terms:

- **MUSICCONSTRAINTBENCH benchmark suites:** eight frozen JSONL files containing all 2,180 benchmark items plus the 400-item paraphrase suite. Each item includes the prompt and the constraint specifications; the generation seed is a parameter of the suite sampler rather than a per-row field.
- **Verifier code:** the eight constraint verifiers, the ABC parser, the reward function, the historical verifier, the corrected strict gate, and both sets of scoring reports.
- **SFT and RL data:** the 12,890 supervised training pairs and 6,000 RL prompts.
- **Model outputs:** per-item completions and per-item verifier decisions (with all sub-signals) for the fixed MusicRLVR checkpoint, the SFT checkpoint, all zero-shot baselines, and the diagnostic configurations.
- **Evaluation reports:** aggregate reports, significance tests, quality statistics, prompt-overlap and unique-prompt analyses, strict-gate audits, and the training logs from which the reward curves are drawn.
- **Asset generation scripts:** `make_assets.py` for the initial tables, `make_audit_assets.py` for supplementary analyses, the final reporting pass `make_checkpoint_assets.py -sync`, and one `make_fig_*.py` per figure, all reading saved reports, so every number in the paper is traceable to a specific evaluation output.

The IrishMAN corpus (Wu et al., 2023) used for SFT training is publicly available; source-specific rights and attribution should be preserved for corpus-derived targets. Model distribution follows the licence of the exact Qwen3-4B-Instruct checkpoint (Yang et al., 2025). No human subjects were involved in this research; all evaluation is automated.

C BENCHMARK AND VERIFIER DETAILS

C.1 VERIFIER IMPLEMENTATION DETAILS

The historical gate ρ , used in training and all tables labelled *A*, runs first. It extracts the ABC block from the completion and rejects the output if extraction fails, if the text contains `| :` or `: |`, if `music21` raises or exceeds a 20-second parse timeout, or if the selected first part has fewer than eight notes, fewer than four non-empty measures, or any chord object. It does not re-check every formatting instruction printed in the prompt. Outputs that clear the gate are then scored by the family verifiers.

Because these checks read the selected first part, extra voices, wrappers and some missing headers clear this gate. Appendix C.2 reports the corrected checks on the same saved completions.

Each of the eight constraint families is implemented as a deterministic function that takes a `music21 Score` object and the constraint parameters, returning a binary verdict. We describe the non-trivial implementation choices below.

key. As discussed in §3.2, the verifier uses a hybrid policy: the **K**: header must match the request, *and* key-profile analysis of the note content (`score.analyze("key")`, which in `music21 10.5.0` selects the Aarden–Essen profiles) must return the requested key or its relative major/minor. On a calibration set of 400 IrishMAN tunes, this hybrid policy achieves 0.890 agreement with the corpus key annotation, compared to 0.750 for content analysis alone and 1.000 (trivially) for header-only checking.

meter. The verifier checks two conditions: (1) the time signature in the **M**: header matches the request, and (2) every non-empty bar’s total duration (notes and rests, in quarter-note equivalents) equals that signature’s bar length within 10^{-4} . Incomplete initial or final bars receive no tolerance.

length. The number of non-empty bars is counted after parsing and compared with the request. The prompt’s “no pickup bar” is enforced here as well: the first and last written bar must each carry a full measure of the tune’s own time signature, so an output that opens with an anacrusis and closes with its complement fails even when no `meter` constraint is requested.

range. Every sounding pitch is compared against the specified minimum and maximum (as MIDI numbers). Rests are ignored.

final. The pitch class of the last sounding note is compared against the request; octave is always ignored.

rhythm. Every note’s duration in quarter-note units, rounded to four decimals, must belong to the admissible set, and any rest fails the check. A dotted or tied value therefore passes only if its parsed duration is itself admissible.

leap. Every pair of consecutive sounding pitches is examined; the verifier returns 0 if any interval exceeds the stated maximum (in semitones).

structure. A tune with other than sixteen non-empty bars fails outright; otherwise it is split into four units of four bars each. Each bar is reduced to its signature (event type, MIDI pitch and duration rounded to four decimals, rests included). The minimal pair sets are AABA: $E = \{(1, 2), (1, 4)\}$, $D = \{(1, 3)\}$; AABB: $E = \{(1, 2), (3, 4)\}$, $D = \{(1, 3)\}$; ABAB: $E = \{(1, 3), (2, 4)\}$, $D = \{(1, 2)\}$. For each pair in E (Equation 2) the verifier compares the bar signatures position by position; for each pair in D it checks that at least one position differs. The graded training signal $\bar{e}$ is the fraction of bar positions that match across all pairs in E ; the binary evaluation signal requires all to match.

C.2 STRICTER FIXED-INSTRUCTION GATE AUDIT

This audit covers the diagnostic checkpoints of Table 33 and the three matched RL checkpoints of Table 2.

The historical gate can accept a second voice because most checks inspect only the first part, whereas key analysis inspects the whole score. It also does not reject every wrapper, missing header, lyric or grace note. We retain that verifier as `verifiers_legacy.py` to reproduce the saved training runs, and implement a stricter gate in `verifiers.py`. It requires a single tune with the specified headers, no unconsumed wrapper text, no repeat notation, lyrics or grace notes, and no multiple parts, multiple voices, chord objects or overlapping notes. Whitespace is normalised when detecting wrappers. The eight family checks remain unchanged. This hardens the fixed-instruction checks; conformance to the full ABC standard and musical correctness are outside what it verifies.

The stricter metric is

$$A_{\text{strict}}(y, C) = A(y, C) \mathbb{I}[g_{\text{strict}}(y)].$$

`audit_strict_gate.py` evaluates the additional gate on saved completions; no regeneration or retraining is involved. For the configurations within this manuscript’s scope, the audit covers 175 evaluation files and 43920 rows: the three matched checkpoints of Table 2 on all eight suites, plus the baseline, training-configuration, ablation, sweep, paraphrase and retained dynamics checkpoints. All file-level outcomes, rejection reasons and source hashes are saved separately. Tables 14–15 show the original six suites; Table 16 adds Dense and Edge. Across the audit subset, 201 previously successful rows fail the stricter gate, and *none of them belongs to the three matched RL rows of Table 2*: those three checkpoints incur no additional gate rejection and lose no success on any of the 6,540 items they were scored on, so every rate in those three RL rows is unchanged under A_{strict} . The main comparison and reward curves use A ; this audit measures the additional fixed-format requirements, which are not included in the main score.

Table 14: All-satisfied rate under the stricter fixed-instruction gate A_{strict} on the original-suite completions. The last three rows are the matched checkpoints of Table 2; every one of their rates equals the corresponding A rate.

Model	Mixed	Single	Seen	Unseen	High- k	Structure
Base 4B	0.160	0.225	0.100	0.040	0.037	0.100
Qwen3-8B	0.120	0.210	0.115	0.050	0.006	0.000
Qwen3-32B	0.357	0.535	0.240	0.190	0.094	0.267
Llama-70B	0.377	0.695	0.300	0.190	0.050	0.167
MiniMax-M3	0.367	0.580	0.315	0.190	0.100	0.008
ChatMusician	0.000	0.000	0.000	0.000	0.000	0.000
SFT	0.477	0.745	0.415	0.440	0.231	0.600
MusicRLVR (diagnostic reference)	0.663	0.835	0.635	0.595	0.356	0.808
Direct GRPO ($R = A$)	0.797	0.960	0.745	0.565	0.469	0.908
MusicRLVR (Direct)	0.807	0.985	0.815	0.595	0.575	0.967
MusicRLVR (SFT init.)	0.577	0.880	0.585	0.530	0.350	0.383

Table 15: Additional gate rejections across the six original suites (1,180 rows per model). Lost successes previously had $A = 1$ and now have $A_{\text{strict}} = 0$; an extra rejection need not change success if a family already failed. Reasons list only the first detected violation per output.

Model	New rejects	Lost successes	Leading reasons
Base 4B	7	0	non abc wrapper or unconsumed text: 7
Qwen3-8B	6	1	non abc wrapper or unconsumed text: 5, grace notes forbidden: 1
Qwen3-32B	22	4	non abc wrapper or unconsumed text: 19, repeat signs forbidden: 2
Llama-70B	26	3	non abc wrapper or unconsumed text: 26
MiniMax-M3	1	0	non abc wrapper or unconsumed text: 1
ChatMusician	749	151	missing required header: 734, non abc wrapper or unconsumed text: 15
SFT	0	0	–
MusicRLVR (diagnostic reference)	0	0	–
Direct GRPO ($R = A$)	0	0	–
MusicRLVR (Direct)	0	0	–
MusicRLVR (SFT init.)	0	0	–

Table 16: Historical and strict all-satisfied rates on the two additional suites, using the same saved completions.

Model	Dense A	Dense A_{strict}	Edge A	Edge A_{strict}
Base 4B	0.008	0.008	0.010	0.010
Qwen3-8B	0.020	0.020	0.032	0.032
Qwen3-32B	0.074	0.074	0.112	0.102
Llama-70B	0.040	0.038	0.058	0.056
MiniMax-M3	0.088	0.088	0.160	0.160
ChatMusician	0.020	0.000	0.046	0.000
SFT	0.290	0.290	0.076	0.076
MusicRLVR (diagnostic, re-evaluated)	0.370	0.370	0.162	0.162
Direct GRPO ($R = A$)	0.322	0.322	0.300	0.300
MusicRLVR (Direct)	0.362	0.362	0.310	0.310
MusicRLVR (SFT init.)	0.366	0.366	0.134	0.134

The SFT and fixed SFT-to-GRPO checkpoints incur no additional gate rejections on any of the eight suites. ChatMusician’s strict success rates are zero because its otherwise accepted outputs omit required headers or include wrappers. This audit therefore changes some baseline scores; a strict-format failure is a formatting verdict.

The corrected gate has regression checks for valid monophonic output and for multiple voices, chords, grace notes, lyrics, missing title, prose wrappers and repeat signs. Historical training uses the explicitly frozen gate; any future strict-gate training should be reported as a new experiment.

C.3 KEY VERIFIER CALIBRATION

The hybrid key verifier (§3.2) is the most complex verifier in our suite, involving both header matching and content-based key analysis. We provide the full calibration data here.

Calibration dataset. We sampled 400 tunes from the IrishMAN corpus at random (seeded), each with a key annotation in the K: header. We then ran four key-verification strategies and measured their agreement with that annotation (Table 17):

Table 17: Key verifier calibration on 400 IrishMAN tunes. Agreement is the fraction of tunes where the verifier’s answer matches the corpus key annotation.

Strategy	Agreement
Header only (trivially gameable)	1.000
Content analysis only (Aarden–Essen profiles)	0.750
Content analysis with relative-key tolerance	0.890
Hybrid: header AND content $\in \{\text{key, relative}\}$	0.890

The reference here is the corpus K: annotation, which the header-only policy reproduces by construction. The hybrid policy matches the content-with-tolerance policy on this set; its advantage appears when the model might game the header, since a policy that prints K:G above music in C major passes the header-only check but fails the hybrid one. The remaining 0.110 is disagreement between the analysis routine and that annotation, which modal tunes, extensive chromaticism and very short tunes can each move. The same policy applies to every model we score. This is agreement with source metadata, not validation against independent positive and negative examples of musical key. The verifier gives an operational target and explicitly tolerates relative major/minor; determinism does not make it an error-free musicological ground truth.

Effect on reported numbers. The key family’s per-constraint satisfaction rate (Table 38) is 0.862 on the single split and 0.671 on the high- k split. If we used a header-only verifier, these would be higher (the model rarely writes the wrong K: header), but header-only acceptance does not inspect the notes. Content analysis with relative-key tolerance carries the 0.110 disagreement above; exact content-only matching disagrees on 0.250 of the calibration tunes. We adopt the hybrid policy and log all sub-signals, so future work can re-score under any policy.

C.4 HELD-OUT PAIR RATIONALE

The four held-out pairs (key, rhythm), (meter, leap), (final, structure) and (length, range) were chosen to form a *perfect matching* on the eight families, ensuring that (1) every family participates in exactly one held-out pair, (2) no family appears in multiple exclusions, without implying equal sampled frequency or equal difficulty, and (3) the matching constrains the maximum testable k to 4 without touching any held-out pair, which naturally defines the high- k split.

The specific pairing was selected to maximise the diversity of inter-family interaction types:

- **key–rhythm:** one harmonic, one temporal, with no obvious interaction (a tune can be in any key with any rhythm vocabulary).
- **meter–leap:** one temporal, one melodic; the meter determines bar duration but should not affect interval size.

- **final-structure**: one local (last note), one global (16-bar form); the most interesting test of whether a model can coordinate endpoint and large-scale structure without having seen them together.
- **length-range**: both are “bounding” constraints (number of bars, pitch boundaries) that interact primarily through the amount of melodic material available.

The supervised dataset contains zero instances of any held-out pair, verified by an exhaustive check over all 12,890 pairs before training.

C.5 ABC NOTATION PRIMER

ABC notation is a plain-text format for monophonic music, widely used in folk-music archives. A minimal valid tune contains a header block and a body:

```
X:1           % reference number
T:My Tune    % title
M:4/4        % meter (time signature)
L:1/8        % default note length
K:G          % key signature
G2 BG dGBG | G2 Bd cAFA | G2 BG dGBG | cABG AGFG |
```

Key conventions: note names C D E F G A B denote the octave starting at middle C; lowercase c d e f g a b denote one octave higher; trailing apostrophes (e.g. c') raise further. A digit after a note name multiplies its duration by that factor relative to the default length L:. Thus G2 in a tune with L:1/8 is a quarter note on G. The pipe character | separates bars.

Because ABC is plain text, a language model generates it token by token in exactly the way it generates code or prose. The `music21` library (Cuthbert & Ariza, 2010) parses ABC into a rich internal representation (pitch, duration, key analysis, interval computation) that our verifiers operate on. If parsing fails, for instance because the model outputs prose instead of notation or uses an invalid header like L:8 instead of L:1/8, the gate returns $\rho(y) = \perp$, so $A(y, C) = 0$ (Equation 1).

C.6 ABC NOTATION: A WORKED EXAMPLE

To help readers unfamiliar with ABC notation, we provide a token-by-token walkthrough of a model-generated output that passes all four constraints of a high- k problem.

Problem. Meter 3/4; exactly 8 bars; last note E; eighth and quarter notes only.

Model output:

```
X:17011
T:Tune
L:1/8
M:3/4
K:Edor
B2 B2 AG | E2 E2 DE | F2 F2 ED | B2 B2 AG |
E2 E2 DE | F2 F2 ED | B2 B2 AG | E2 E2 DE |
```

Token-by-token analysis:

- **X:17011**: Reference number (arbitrary, used by databases).
- **T:Tune**: Title (arbitrary, not constrained).
- **L:1/8**: Default note length is an eighth note. This means B alone is an eighth note, B2 is two eighth notes = one quarter note.
- **M:3/4**: Time signature: three quarter notes per bar. Since L:1/8, each bar must contain six eighth-note equivalents. ✓ meter.
- **K:Edor**: Key of E Dorian (E minor with raised sixth). The key constraint is not active in this problem, so any key is acceptable.

- B2 B2 AG: Bar 1. B2 = quarter note B (two eighths); B2 = quarter note B; A = eighth note A; G = eighth note G. Total: $2 + 2 + 1 + 1 = 6$ eighth-note equivalents. ✓ meter. Durations used: quarter, eighth. ✓ rhythm.
- |: Bar line separator.
- E2 E2 DE: Bar 2. Same structure. Total 6. ✓
- ... (bars 3–7 follow the same pattern) ...
- E2 E2 DE |: Bar 8 (last bar). The final token is E, an eighth note on pitch E. ✓ final.

Verification summary: meter ✓ (all 8 bars have 6 eighth-note equivalents); length ✓ (exactly 8 bars); final ✓ (last note is E); rhythm ✓ (only eighth and quarter notes appear). All four constraints satisfied; $A = 1$.

Note that the tune reuses three bar motifs in the order A–B–C–A–B–C–A–B, the minimal-motif style examined in §5.6. The tune is *correct* under every stated requirement; perceptual quality is outside what the verifiers measure.

C.7 CONSTRAINT RENDERING EXAMPLES

Table 19 shows the complete set of constraint-text renderers: for each family, the table lists the parameter types and one representative rendering. These renderers are deterministic functions, so every benchmark item with the same family and parameters produces the same constraint text.

The constraint texts are designed to be unambiguous to a musically literate reader. The `meter` renderer explicitly states “every bar must contain exactly the right total duration” because simply saying “the meter is 3/4” could be interpreted as a header-only requirement. Similarly, the `structure` renderer spells out the matching/differing requirement rather than assuming the model knows what AABA form means.

Table 18: Table 1 with the prompt column restored. Under each family name is the span it is checked over; the second column lists the default sampler’s values (the verifier domain Θ_f is wider), the third the sentence the prompt template emits for one such value, verbatim from the renderer. **structure** is set apart as the only cross-section form family.

Family	Default sampler values	Prompt line (one sampled value)
key global	13 keys: 8 major, 5 minor	<i>the tune must be in D minor</i>
meter every bar	4/4, 3/4, 6/8, 2/4; each bar must sum to the signature	<i>the meter must be 6/8, and every bar must contain exactly the right total duration</i>
length global	8, 12 or 16 bars, no pickup	<i>the tune must be exactly 16 bars long (no pickup bar)</i>
range every note	4 inclusive pitch spans, e.g. G3–A5	<i>every note must lie between G3 and A5 inclusive</i>
final last note	a pitch class, any octave (a scale degree 1/3/5 when a key is present)	<i>the last note of the tune must be E (any octave)</i>
rhythm every note	4 admissible duration sets; no dotted values, no rests	<i>use only eighth and quarter notes (no dotted notes, no other durations, no rests)</i>
leap adjacent notes	largest melodic interval: 7, 9 or 12 semitones	<i>melodic motion must be smooth: no melodic leap larger than a major sixth</i>
structure 4-bar units	AABA, AABB or ABAB over 16 bars; equal letters identical, different letters distinct	<i>the tune must be exactly 16 bars with an ABAB form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ</i>

Table 19: Constraint rendering for all eight families. The “Parameters” column shows the parameter space; the “Rendered text” column shows the template output for one example value. Paraphrase variants (Appendix C.8) replace these templates with alternative wordings.

Family	Parameters	Rendered text (example)
key	key name $\times$ mode	“The tune must be in G major.”
meter	time signature	“The meter must be 3/4, and every bar must contain exactly the right total duration.”
length	bar count	“The tune must be exactly 16 bars long (no pickup bar).”
range	low pitch, high pitch	“Every note must lie between C4 and C6 inclusive.”
final	pitch class; any octave	“The last note of the tune must be D (any octave).”
rhythm	admissible duration set	“Use only eighth and quarter notes (no dotted notes, no other durations, no rests).”
leap	max interval name	“Melodic motion must be smooth: no melodic leap larger than a perfect fifth.”
structure	form label, unit size	“The tune must be exactly 16 bars with an AABA form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ.”

C.8 PARAPHRASE EXAMPLES

The paraphrase suite re-renders the same constraint values with alternative wordings. Table 20 shows three examples, illustrating the range of surface variation; the constraint parameters are unchanged.

D ADDITIONAL EXPERIMENTS

Appendix D.5 contains the matched comparison. All other RL analyses in this section preserve the diagnostic study, with SFT initialisation unless a Direct row or another configuration is explicitly named. Table 33 collects those historical reference scores.

Unless explicitly labelled otherwise, all-satisfied rates use A (Equation 1), as in the main comparison; per-family statistics report the individual verifier verdicts. Appendix C.2 separately reports the additional format-gate audit.

Table 20: Paraphrase examples. Each row shows the original constraint text and one paraphrased variant; the constraint parameters are identical.

Original	Paraphrased
The tune must be in A minor.	Compose the melody in the key of A minor.
The meter must be 4/4, and every bar must contain exactly the right total duration.	Write it in 4/4 time, and make every measure add up to exactly the right duration.
Melodic motion must be smooth: no melodic leap larger than an octave.	Keep the melodic motion conjunct: no interval between consecutive notes may exceed 12 semi-tones.

D.1 PARAPHRASE AND TRAINING DYNAMICS

Table 21 re-renders the single and seen-combination problems with alternative wordings of identical constraint values, and Table 22 gives the numeric values behind Figure 5(d).

Table 21: Diagnostic study: instruction paraphrase, scored by A (the format audit leaves these verdicts unchanged). The original column averages the single and seen-combination splits; the paraphrased column re-renders the same 400 problems with alternative wordings of identical constraint values.

Model	Original wording	Paraphrased	Δ
SFT	0.580	0.535	-0.045
MUSICRLVR	0.735	0.688	-0.047

Table 22: Diagnostic study: numeric values behind Figure 5(d). Curve rows come from a full-retention re-run of the SFT-initialised reference configuration; “original run” gives the three checkpoints the original run retained. Its step-300 value 0.660 is a separate evaluation of the weights Table 3 scores at 0.663, within the re-evaluation shift of Appendix B.9; the two are reported separately, not pooled. The last row uses the symmetric structure reward of Appendix E.2; it and the symmetric row of Table 3 (0.590 on Mixed) are that reward’s two separate runs, not two evaluations of one checkpoint.

Step	50	100	150	200	250	300
All satisfied	0.570	0.593	0.633	0.687	0.693	0.703
meter	0.716	0.852	0.901	0.926	0.914	0.926
final	0.505	0.505	0.558	0.642	0.632	0.632
key	0.838	0.838	0.787	0.838	0.800	0.812
structure	0.767	0.836	0.808	0.740	0.822	0.781
All satisfied (original run, retained ckpts)	–	–	–	0.630	0.640	0.660
All satisfied (symmetric structure reward)	0.477	0.463	0.463	0.530	0.537	0.553

D.2 REJECTION SAMPLING AND A MUSIC-SPECIALISED BASELINE

Table 4(b) spends the verifier at inference time instead: for each item we draw eight completions, matching the GRPO group size $G = 8$ and the rollout temperature (1.0), from both trained models, so selection sees the same per-prompt candidate count as one GRPO group, and report $\text{pass}@k$ under the unbiased estimator $(1 - \binom{n-c}{k}) / \binom{n}{k}$ for c of n samples satisfying all constraints, averaged over items). Selection helps both models and MUSICRLVR stays ahead at every matched candidate budget, on both suites, so verifier-guided selection stacks with training rather than substituting for it; note that, unlike the trained policy alone, best-of- N requires the verifier and N generations and N verifications at deployment.

ChatMusician-7B (Yuan et al., 2024) is evaluated zero-shot under the same protocol as every other untrained model, through its released Human:Assistant: template and greedy decoding (context 2,048 tokens). Under A , it reaches 0.160 on the mixed suite and 0.000 on the structure split, where its parse rate is 0.158. Under the additional format gate its score is zero on all eight suites (Tables 14 and 16), mainly because required headers are missing. Trained on corpus ABC, it writes repeat signs on most form prompts, which the instruction forbids and the parse gate rejects. Its training objective is stylistic generation rather than stated requirements, so the protocol is not matched to it. This measurement bounds this model under this template, context budget and decoding setting; it

does not establish what music-specialised pre-training confers in general, which would need a prompt format and budget tuned to that model.

D.3 EDGE SUBSETS AND A STRICTER KEY CRITERION

Both analyses use saved per-item verdicts, without regeneration or retraining. `src/rescore_subsets.py` computes the key-sensitivity audit under the historical gate; the final `make_checkpoint_assets.py` pass uses the same A verdicts for Edge subsets.

Edge mixes two factors. Table 23 splits it by whether an item requests one of the four held-out pairs. On the 329 items without a held-out family pair, MusicRLVR reaches 0.210 against 0.100 for the warm start (exact McNemar $p = 2.03 \times 10^{-6}$) and 0.195 for MiniMax-M3 ($p = 0.685$); on the 171 items that do request a held-out pair the order reverses (0.070 against 0.094). The margin over the warm start is wider on the pair-free subset than on Edge as a whole.

Table 23: Diagnostic study: Edge split by held-out-pair membership, using the main-table success criterion. Pair-free items contain none of the four held-out family pairs.

Model	All ($n = 500$)	Pair-free ($n = 329$)	With held-out pair ($n = 171$)
Base 4B	0.010	0.012	0.006
Qwen3-8B	0.032	0.049	0.000
Qwen3-32B	0.112	0.125	0.088
Llama-70B	0.058	0.067	0.041
MiniMax-M3	0.160	0.195	0.094
ChatMusician	0.046	0.055	0.029
SFT	0.076	0.100	0.029
MUSICRLVR (re-evaluated)	0.162	0.210	0.070

The deployed key check requires the header to match and the analyzed key to be the requested key or its relative (§C.3). Tables 24 and 25 hold the historical format gate fixed and report every rate with that tolerance and without it, that is with key satisfied only when header and analysis both give the requested key. Rates stay unchanged or decrease. The largest baseline decrease is 0.060 (Qwen3-32B on Unseen); the largest decrease for the fixed MusicRLVR checkpoint is 0.030 on Seen. The SFT-initialised route remains highest-scoring among the models in this audit on seven splits; on Edge, the two leading rates stay within one item of each other (0.150 against 0.152).

Table 24: Diagnostic study: All-satisfied rate with the relative-key tolerance and without it (hybrid/exact key), six original suites, holding the historical format gate fixed. Baselines show the paired rates; the two MusicRLVR rows report the fixed-checkpoint rate under each key rule.

Model / key rule	Mixed	Single	Seen	Unseen	High- k	Structure
Base 4B	0.160/0.157	0.225/0.210	0.100/0.090	0.040/0.030	0.037/0.031	0.100/0.100
Qwen3-8B	0.123/0.117	0.210/0.200	0.115/0.110	0.050/0.050	0.006/0.006	0.000/0.000
Qwen3-32B	0.363/0.340	0.535/0.485	0.250/0.240	0.190/0.130	0.094/0.062	0.267/0.267
Llama-70B	0.380/0.370	0.695/0.675	0.300/0.295	0.195/0.185	0.056/0.056	0.167/0.167
MiniMax-M3	0.367/0.363	0.580/0.550	0.315/0.310	0.190/0.185	0.100/0.100	0.008/0.008
ChatMusician	0.160/0.147	0.290/0.280	0.130/0.120	0.090/0.085	0.006/0.000	0.000/0.000
SFT	0.477/0.453	0.745/0.715	0.415/0.385	0.440/0.415	0.231/0.225	0.600/0.592
MusicRLVR (hybrid)	0.663	0.835	0.635	0.595	0.356	0.808
MusicRLVR (strict key)	0.637	0.815	0.605	0.580	0.344	0.783

Table 25: Diagnostic study: Hybrid/exact key on Dense and Edge, holding the historical format gate fixed; MusicRLVR uses the single re-evaluation.

Model	Dense	Edge
Base 4B	0.008/0.006	0.010/0.008
Qwen3-8B	0.020/0.018	0.032/0.032
Qwen3-32B	0.074/0.056	0.112/0.102
Llama-70B	0.040/0.032	0.058/0.048
MiniMax-M3	0.088/0.084	0.160/0.152
ChatMusician	0.020/0.014	0.046/0.042
SFT	0.290/0.266	0.076/0.070
MUSICRLVR (re-evaluated)	0.370/0.342	0.162/0.150

D.4 SIGNIFICANCE TESTS

The matched Direct-versus-SFT-initialised tests are in Appendix D.5; the tables below cover the diagnostic reward-design comparisons.

Table 26 reports exact McNemar tests for the single-run diagnostic reward-design experiment under the main-table success criterion, pairing the same item IDs. These tests condition on the specified checkpoints; Appendix D.7 lists the success counts underlying the historical reference rates. Table 27 reports the Edge comparison. All p values are nominal pairwise tests.

Table 26: Diagnostic study: Single-run reward-design comparisons (nominal exact McNemar p) over per-item outcomes.

Comparison	Mixed	Single	Seen	Unseen	High- k	Struct.
Reference run vs. SFT	$< 10^{-4}$	0.002	$< 10^{-4}$	$< 10^{-4}$	0.007	$< 10^{-4}$
Reference run vs. binary credit, 21% share	0.076	0.597	1.000	0.312	1.000	$< 10^{-4}$
Reference run vs. binary credit, 33% share	0.822	0.583	0.625	0.771	0.678	$< 10^{-4}$

Table 27: Diagnostic study: Paired comparison on Edge. “Only RL” and “Only baseline” count discordant items in the designated single evaluation of each model.

Comparison	Only RL	Only baseline	Exact p	Direction
Edge: RL vs. MiniMax	62	61	1.000000	No detected difference

D.5 MATCHED COMPARISON

This study supplies the three RL rows in Table 2. All runs use seed 43, checkpoint 300, the same 6,000-prompt training pool and 48 completions per update. Two policy accelerators use micro-batch 2 and accumulation 12; a third serves rollouts with memory fraction 0.92. The remaining GRPO settings match Table 5. Direct binary uses $\lambda = 0$ ($R = A$), and both MusicRLVR routes use $\lambda = 0.7$. The reward weight is the only changed training setting between the two Direct arms. The SFT-initialised arm changes both the initial policy and its fixed KL reference. All evaluations use the same item IDs and the all-satisfied criterion A .

Table 28: Success counts and rates for the matched final checkpoints.

Split	Direct, binary	Direct, MusicRLVR	SFT init., MusicRLVR
Mixed	239/300 (0.797)	242/300 (0.807)	173/300 (0.577)
Single	192/200 (0.960)	197/200 (0.985)	176/200 (0.880)
Seen	149/200 (0.745)	163/200 (0.815)	117/200 (0.585)
Unseen	113/200 (0.565)	119/200 (0.595)	106/200 (0.530)
High- k	75/160 (0.469)	92/160 (0.575)	56/160 (0.350)
Structure	109/120 (0.908)	116/120 (0.967)	46/120 (0.383)
Dense	161/500 (0.322)	181/500 (0.362)	183/500 (0.366)
Edge	150/500 (0.300)	155/500 (0.310)	67/500 (0.134)

Pooled comparison. The eight suites are disjoint item sets whose union is the 2,180 items of MUSICCONSTRAINTBENCH, so the same paired test applies to the union. Against binary reward, Direct MusicRLVR satisfies 1265 items against 1188 (0.580 against 0.545), winning 229 and losing 152 of the 381 items on which the two disagree (exact two-sided $p = 9.39 \times 10^{-5}$). Against the SFT-initialised route it satisfies 1265 against 924, winning 554 and losing 213 ($p < 10^{-4}$); the binary-reward arm also leads that route, 1188 against 924 ($p < 10^{-4}$).

Per-suite breakdown. Graded credit has the higher rate on all eight suites, though on Mixed and Edge the margin is 0.010, within the re-evaluation spread of Table 13. Table 31 applies the eight-suite Holm correction inside each comparison: under it the reward-granularity differences are individually unresolved at these discordant counts, while six of the eight initialisation differences are significant, the exceptions being Unseen and Dense. The initialisation comparison uses the identical protocol, with Direct ahead on seven suites and behind on Dense by two successes. All three models pass

Table 29: Parse-gate pass rates on the six original suites, for every model of Table 2. Each trained 4B checkpoint clears the gate on every item it was given.

Model	Mixed	Single	Seen	Unseen	High- k	Structure
Qwen3-4B	0.513	0.450	0.510	0.545	0.362	0.925
Qwen3-8B	0.367	0.465	0.345	0.305	0.219	0.158
Qwen3-32B	0.973	1.000	0.960	0.945	0.938	0.983
Llama-3.1-70B	0.970	1.000	0.980	0.980	0.944	1.000
MiniMax-M3	0.577	0.670	0.565	0.465	0.531	0.083
ChatMusician-7B	0.693	0.800	0.680	0.670	0.575	0.158
SFT (4B)	1.000	1.000	1.000	1.000	1.000	1.000
Direct GRPO ($R = A$)	1.000	1.000	1.000	1.000	1.000	1.000
MusicRLVR (SFT init.)	1.000	1.000	1.000	1.000	1.000	1.000
MusicRLVR (Direct)	1.000	1.000	1.000	1.000	1.000	1.000

Table 30: Parse-gate pass rates for the matched evaluations.

Model	Mixed	Single	Seen	Unseen	High- k	Structure	Dense	Edge
Direct GRPO ($R = A$)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MusicRLVR (SFT init.)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MusicRLVR (Direct)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.992

the parse gate on every original-suite item; on Edge the Direct MusicRLVR model passes 496/500, while the binary-reward and SFT-initialised models pass 500/500 (Tables 29 and 30).

What the residual failures look like. Table 32 decomposes the 58 Mixed items the Direct MusicRLVR checkpoint does not satisfy, from the same saved per-item verdicts as Table 28. The failures are shallow rather than pervasive: 48 of the 58 miss a single family and only 2 miss three, none fails the parse gate, and **structure** accounts for both the largest per-family deficit (22 of the 73 items requesting it) and the most common sole miss (18 items). This is the matched checkpoint of Table 2; the diagnostic split-level patterns of Appendix E.1 are a separate checkpoint and are not comparable item for item.

Table 31: Exact two-sided McNemar tests for the matched Direct MusicRLVR checkpoint against each control; discordant counts are successes by only one checkpoint, and Holm correction is applied over the eight suites within each comparison. These tests condition on the saved completions and checkpoints, and do not estimate variation across training runs or repeated decoding.

Comparison	Split	Only Direct	Only control	Exact p	Holm p
Binary reward	Mixed	19	16	0.7359	1.0000
	Single	5	0	0.0625	0.3125
	Seen	25	11	0.0288	0.1729
	Unseen	21	15	0.4050	1.0000
	High- k	32	15	0.0186	0.1304
	Structure	7	0	0.0156	0.1250
	Dense	78	58	0.1029	0.4117
	Edge	42	37	0.6530	1.0000
SFT init.	Mixed	90	21	$< 10^{-4}$	$< 10^{-4}$
	Single	22	1	$< 10^{-4}$	$< 10^{-4}$
	Seen	65	19	$< 10^{-4}$	$< 10^{-4}$
	Unseen	36	23	0.1175	0.2350
	High- k	58	22	$< 10^{-4}$	0.0002
	Structure	71	1	$< 10^{-4}$	$< 10^{-4}$
	Dense	90	92	0.9409	0.9409
	Edge	122	34	$< 10^{-4}$	$< 10^{-4}$

Table 32: Matched study: the 58 unsatisfied Mixed items of the Direct MusicRLVR checkpoint. “Missed” counts items requesting a family whose verifier fails; “sole miss” restricts to the 48 items that miss exactly one family. Families are ordered by missed count; *rhythm* is omitted, with no miss among the items requesting it.

Family	Requested	Missed	Sole miss
structure	73	22	18
length	65	14	10
key	80	11	9
final	95	11	7
leap	89	7	4
meter	81	3	0
range	80	2	0

Items by number of families missed: 1: 48, 2: 8, 3: 2; parse-gate failures: 0.

D.6 DIAGNOSTIC REFERENCE RESULTS

The following table gives the diagnostic evaluations. Unless a diagnostic explicitly states another setting, its RL results refer to the SFT-initialised checkpoint in this table. The Direct route is reported only from the matched runs of Table 2, so no Direct row appears here.

Table 33: Diagnostic reference results (SFT-initialised route): all-satisfied rates above, parse-gate pass rates below. A different RL checkpoint from the matched comparison of Table 2. Colours rank rows within this table; missing entries were not evaluated.

Model	Six original splits						Later splits	
	Mixed	Single	Seen	Unseen	High- k	Structure	Dense	Edge
<i>Zero-shot</i>								
Qwen3-4B	0.160	0.225	0.100	0.040	0.037	0.100	0.008	0.010
Qwen3-8B	0.123	0.210	0.115	0.050	0.006	0.000	0.020	0.032
Qwen3-32B	0.363	0.535	0.250	0.190	0.094	0.267	0.074	0.112
Llama-3.1-70B	0.380	0.695	0.300	0.195	0.056	0.167	0.040	0.058
MiniMax-M3	0.367	0.580	0.315	0.190	0.100	0.008	0.088	0.160
ChatMusician-7B	0.160	0.290	0.130	0.090	0.006	0.000	0.020	0.046
<i>Trained (4B)</i>								
SFT (4B)	0.477	0.745	0.415	0.440	0.231	0.600	0.290	0.076
MusicRLVR (SFT init.)	0.663	0.835	0.635	0.595	0.356	0.808	0.370	0.162
<i>Parse-gate pass rate (six original splits)</i>								
Model	Mixed	Single	Seen	Unseen	High- k	Structure		
Qwen3-4B	0.513	0.450	0.510	0.545	0.362	0.925		
Qwen3-8B	0.367	0.465	0.345	0.305	0.219	0.158		
Qwen3-32B	0.973	1.000	0.960	0.945	0.938	0.983		
Llama-3.1-70B	0.970	1.000	0.980	0.980	0.944	1.000		
MiniMax-M3	0.577	0.670	0.565	0.465	0.531	0.083		
ChatMusician-7B	0.693	0.800	0.680	0.670	0.575	0.158		
SFT (4B)	1.000	1.000	1.000	1.000	1.000	1.000		
MusicRLVR (SFT init.)	1.000	1.000	1.000	1.000	1.000	1.000		

D.7 FIXED-CHECKPOINT EVALUATION

The diagnostic reference checkpoint is `outputs/grpo_4b_v2/checkpoint-300`. Table 34 gives its success counts on the six original suites; on Dense and Edge it reaches 185/500 and 81/500. Its first evaluation supplies the original-suite rates; the separate re-evaluation is documented in Appendix B.9. Success counts for the three matched checkpoints are in Table 28.

Table 34: Diagnostic study: success counts and all-satisfied rates for the SFT-initialised reference checkpoint on the six original suites, matching Table 33.

Split	SFT init. count	Rate
Mixed	199/300	0.663
Single	167/200	0.835
Seen	127/200	0.635
Unseen	119/200	0.595
High- k	57/160	0.356
Structure	97/120	0.808

D.8 HYPER-PARAMETER SWEEPS

Table 35 lists the values plotted in Figure 5(b). Each cell is one training run per setting with a single parameter moved from the SFT-initialised reference configuration, scored on Mixed and Structure. The reward mixing weight is swept the same way in Appendix D.9.

Scope of the sensitivity evidence. Learning rate has the largest observed Mixed spread (0.270), compared with 0.063 for KL coefficient and 0.123 for group size. Every setting exceeds the SFT Mixed baseline. Each cell is one training run, so these ranges describe observed sensitivity across the tested settings. Structure spans 0.458–0.833 across the three, without a monotone pattern along any of them.

The G sweep fixes 48 completions per optimiser step. Accordingly $G = 4, 8, 16$ uses 12, 6, 3 prompt groups per step, for 3,600, 1,800, 900 prompt draws in 300 steps. The corresponding logged epochs are 0.60, 0.30, 0.15. It compares allocation of a fixed completion budget, not group size at fixed prompt exposure.

Table 35: Diagnostic study: All-satisfied rate on the mixed and structure suites for each hyper-parameter setting, one training run each. * marks the SFT-initialised reference configuration. Structure varies substantially across these settings.

Hyper-parameter	Value	Mixed	Structure
Learning rate	5×10^{-7}	0.567	0.808
	1×10^{-6} *	0.663	0.808
	2×10^{-6}	0.710	0.542
	5×10^{-6}	0.837	0.833
KL coefficient β	0	0.607	0.750
	0.01	0.660	0.567
	0.04*	0.663	0.808
	0.1	0.600	0.458
Group size G	4	0.540	0.533
	8*	0.663	0.808
	16	0.577	0.475

D.9 REWARD MIXING WEIGHT

Table 36 moves λ of Equation 4, which weights the per-family average against the binary all-satisfied term, while every other setting stays at the SFT-initialised reference configuration. It is reported separately from Table 35 because that table lists the values plotted in Figure 5(b), and because the form decomposition below is what makes this sweep interpretable.

The default achieves the highest observed scores on both suites among the three weights tested in the single-run sweep. Against $\lambda = 0.7$, the low arm’s Mixed comparison gives exact McNemar $p = 0.368$, while its Structure score is lower ($p = 3.54 \times 10^{-9}$); the high arm scores lower on both ($p = 0.002$ and $p = 4.74 \times 10^{-15}$).

Table 36: Diagnostic study: Reward mixing weight, one training run per setting, scored by the binary criterion of Equation 1. Repetition and contrast count the structure-split items passing all required-equal and all required-distinct unit-pair checks, respectively; * marks the default.

λ	Mixed	Structure	Repetition	Contrast
0.5	0.633	0.425	120/120	55/120
0.7*	0.663	0.808	119/120	104/120
1.0	0.570	0.325	119/120	45/120

Contrast distinguishes the tested weights. The last two columns separate the halves of the form constraint. Repetition is satisfied on 119–120 of the 120 items at every weight, while contrast passes 55 items at $\lambda = 0.5$ and 45 at $\lambda = 1.0$ against 104 at the default: the decrease on Structure accompanies a drop in contrast, not in repetition. The all-satisfied counts behind those rates are 51, 97 and 39; each deficit relative to the contrast count consists of items passing the form check and missing a co-requested family. Outputs whose four units are all identical – which satisfy every required-equal pair and no required-distinct one – rise from 15 items at the default to 65 and 74 at the two ends. The same pattern holds across the rows of Table 3: repetition passes 119–120 items, and the binary form pass count equals the contrast pass count.

Where the weight can act. Write m_i for the per-family average in Equation 4, setting it to zero for a gate-rejected rollout. For $\lambda > 0$, a group in which every rollout has $A(y_i, C) = 0$ has $R(y_i, C) = \lambda m_i$. Equation 5 then gives $\hat{A}_i = (m_i - \bar{m})/(\sigma_m + \epsilon/\lambda)$, where σ_m is the group sample standard deviation of m : the scaling cancels except through the stabiliser. The mixture also controls the reward gap between full and partial success. Consider two gate-passing rollouts for the same prompt: one satisfies all k families; the other fails exactly one family, retaining graded credit $s \in [0, 1)$ for that family. Their reward gap is

$$\Delta R = (1 - \lambda) + \frac{\lambda(1 - s)}{k},$$

which is $(1 - s)/k$ at $\lambda = 1$ and $0.3 + 0.7(1 - s)/k$ at the default. The binary term thus adds an explicit bonus for joint satisfaction; its effect on normalised advantages also depends on the other

Table 37: Diagnostic study: Per-family satisfaction on Mixed: the fraction of items requesting a family whose verifier passes. Δ is the fixed-checkpoint rate minus the SFT rate.

Family	Base	SFT	MusicRLVR	Δ
key	0.237	0.625	0.850	+0.225
meter	0.222	0.630	0.926	+0.296
length	0.015	0.969	1.000	+0.031
range	0.375	0.875	0.975	+0.100
final	0.368	0.421	0.611	+0.189
rhythm	0.416	0.727	0.753	+0.026
leap	0.393	0.764	0.854	+0.090
structure	0.041	0.726	0.644	-0.082

rewards in the group. The sweep supports the default mixture for the tested configuration, with the highest observed compliance and fewest all-identical outputs.

D.10 PER-FAMILY SATISFACTION ON THE ORIGINAL SUITES

Table 33 reports the all-satisfied rate, which is zero whenever any single constraint fails. Table 38 reports each family’s satisfaction rate on each split for the same checkpoint, and Figure 7 gives the same per-family view on Mixed across models.

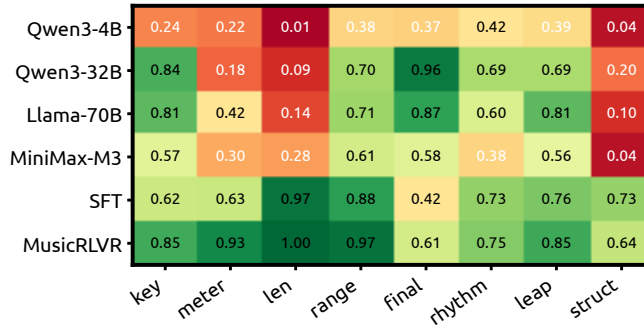


Figure 7: Diagnostic study: Per-family satisfaction on the mixed suite; MusicRLVR cells use the fixed checkpoint (Table 37); structure stays red for every untrained model.

The mixed suite includes 73 structure-family items. Their satisfaction rate is 0.644 for MusicRLVR against 0.726 for SFT. On the dedicated structure split, the all-satisfied rate is 0.808 against 0.600; equal prompt weighting gives 0.637 against 0.522 (Appendix B.7). The splits differ in their requested constraints and prompt weights.

Table 38: Diagnostic study: per-family satisfaction rate for MusicRLVR across the splits. Each cell is the fraction of problems mentioning that family whose own verifier passes at the fixed checkpoint (other constraints may fail); families mentioned by fewer than five problems in a split are marked “-”.

Family	Single	Seen	Unseen	High- k	Structure
final	0.783	0.636	0.407	0.635	-
key	0.862	0.887	0.565	0.671	0.643
leap	0.292	0.935	0.985	0.872	1.000
length	1.000	1.000	1.000	1.000	-
meter	0.812	0.903	0.897	0.890	1.000
range	1.000	0.982	0.955	0.980	1.000
rhythm	0.833	0.790	0.843	0.655	0.692
structure	1.000	0.510	0.667	0.554	0.867

length and range maintain high satisfaction across the splits. The final rate is lower on Unseen than on Single, while key, rhythm and structure also vary with the requested combination. For leap, satisfaction is lower on Single than on the combination splits. Table 38 reports these rates under the corresponding family and parameter mixture.

D.11 PERFORMANCE BY NUMBER OF CONSTRAINTS (k)

Table 39 breaks the mixed suite ($n = 300$; 100/100/60/40 items at $k = 1/2/3/4$) down by the number of constraints, using the same saved diagnostic reference evaluations as Table 33; Figure 1 instead uses the matched runs.

Table 39: Diagnostic study: all-satisfied rate on the mixed suite by number of constraints k . The last column is the $k=1$ rate over the $k=4$ rate; a dagger marks curves not monotone in k (the $k=3$ and $k=4$ cells hold only 60 and 40 items, so one item moves them by 0.017 and 0.025).

Model	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k=1/k=4$
MusicRLVR (SFT init.)	0.850	0.640	0.517	0.475	$1.8\times$
SFT (4B)	0.720	0.430	0.267	0.300	$2.4\times^\dagger$
Qwen3-32B	0.650	0.300	0.133	0.150	$4.3\times^\dagger$
Llama-3.1-70B	0.720	0.340	0.100	0.050	$14.4\times$
MiniMax-M3	0.560	0.360	0.267	0.050	$11.2\times$
Qwen3-4B	0.340	0.110	0.033	0.025	$13.6\times$
Qwen3-8B	0.240	0.120	0.017	0.000	–

The endpoint ratios vary across models. Llama-3.1-70B falls from 0.720 at $k = 1$ to 0.050 at $k = 4$, a $14.4\times$ drop; MiniMax-M3 falls $11.2\times$ and the untrained 4B base $13.6\times$. The SFT-initialised MusicRLVR rates are 0.850 and 0.475, giving the smallest endpoint ratio in the table ($1.8\times$), against $2.4\times$ for SFT. The matched by- k curves are in Figure 1. These slices retain their own family and parameter mixtures.

D.12 OUTPUT LENGTH STATISTICS

Table 40 reports the average number of bars and notes per parsed completion on the mixed suite, supplementing the quality metrics in Table 41.

Table 40: Diagnostic study: Output length in the single-run mixed-suite diagnostic ($n = 300$ problems). Only completions that pass the historical parse gate are included, so the parsed count differs by model; the IrishMAN reference (299 tunes, 19.9 bars) is in Table 41.

Source	Parsed n	Mean bars	Mean notes
Qwen3-4B (zero-shot)	154	29.3	78.4
Qwen3-32B (zero-shot)	292	18.4	58.1
Llama-3.1-70B (zero-shot)	291	21.2	66.1
MiniMax-M3 (zero-shot)	173	17.9	37.0
SFT	300	39.2	167.5
MUSICRLVR	300	28.7	130.3

The supervised model over-generates (39.2 bars against 19.9 for the corpus), and the reinforcement stage pulls this back to 28.7 bars, consistent with a reward that favours concision without an unconditional length term, since a longer tune has more places to violate a constraint. The three larger zero-shot models sit near the corpus length (17.9–21.2 bars; the 4B base averages 29.3), and MiniMax-M3 parses on only 173 of 300 problems.

D.13 LENGTH-CONTROLLED OUTPUT STATISTICS

§5.6 compares the first eight parsed bars of a common item set (Table 4), because the full-tune ratios of Table 41 have length-dependent denominators. This section gives the uncontrolled statistics those ratios come from, and the cross-prompt duplication counts of Table 42. Both read historical-gate outputs, but they measure different things: the first are within-tune proxies, the second counts how often two different prompts produce the same score. Neither measures perceptual quality.

On the full completions, MusicRLVR raises distinct-bar fraction from 0.276 to 0.313 while pitch-class entropy falls $2.00 \rightarrow 1.89$ and 4-gram distinctness $0.294 \rightarrow 0.292$; mean length falls $39.2 \rightarrow 28.7$ bars. The distinct-bar increase here is $0.276 \rightarrow 0.313$; under the length control of §5.6 the same comparison is $0.417 \rightarrow 0.426$ on matched eight-bar prefixes, which is why the matched comparison is the one reported in the main text. These diagnostics describe output variety, not an aesthetic

Table 41: Diagnostic study: distributional statistics of the single-run mixed-suite completions and 299 corpus tunes. No constraint names entropy or either distinctness measure; bar count is constrained on the 138/300 items requesting `length` or `structure`. Mean length differs between rows, so distinctness denominators are unmatched; Table 4 controls for that.

Model	Bars	PC ent.	Dist. bars	Dist. 4-gr.
Corpus	19.9	2.63	0.758	0.676
Base	29.3	2.06	0.366	0.391
SFT	39.2	2.00	0.276	0.294
RL, binary credit	32.2	1.85	0.284	0.289
MUSICRLVR	28.7	1.89	0.313	0.292

Table 42: Diagnostic study: full-score duplication for the first eligible occurrence of each of 220 distinct prompts in that common set, identified by parsed event signatures and ignoring metadata. This measures duplication across prompts, not diversity from repeated sampling of one prompt.

Model	Unique full scores	Most frequent score
SFT	167/220	7/220
MUSICRLVR (diagnostic)	182/220	5/220

advantage of reinforcement learning. Repeated sampling, length-matched full-score comparisons and independent listening evaluations would address different aspects of musical diversity and quality.

D.14 ZERO-VARIANCE GROUPS DURING TRAINING

Equation 5 implies that a group whose G rollouts all receive the same reward contributes no reward gradient. The trainer logs the fraction of such groups per step (`frac_reward_zero_std`); Table 43 summarises it for the two reward variants of the main ablation.

Table 43: Diagnostic study: Fraction of GRPO groups with zero reward variance (duplicate step records from restarts removed), averaged over the first and last quarter of training and over all 300 steps, with the mean within-group reward standard deviation alongside.

Run	Zero-variance groups			Reward std	
	first 25%	last 25%	all	first 25%	last 25%
Binary form credit, original form share (21%)	0.116	0.307	0.221	0.301	0.214
Graded form credit, 33% form share (final)	0.133	0.322	0.254	0.296	0.210

Two readings. In the final configuration about a quarter of all groups carry no reward-driven gradient, rising to a third late in training, which is the quantity the graded term targets. The aggregate should be read together with the prompt mix: the two ablation rows differ in both the reward and the share of structure prompts, and the changed share of form prompts could itself alter the zero-variance fraction.

The matched runs measure the same quantity on two arms that differ only in λ , holding the prompt mix fixed. Averaged over the first 50 updates, binary reward leaves 0.550 of groups with zero reward variance against 0.307 for the graded reward; over the last 50 the two reach 0.823 and 0.823, once most groups succeed outright, and over all 300 updates they average 0.717 and 0.660 (Figure 4(b)). The SFT-initialised arm runs lower throughout (0.120, 0.343, 0.245) while scoring lowest on the benchmark, so the fraction is read between arms that share an initialisation rather than across routes.

D.15 ADDITIONAL TRAINING CURVES

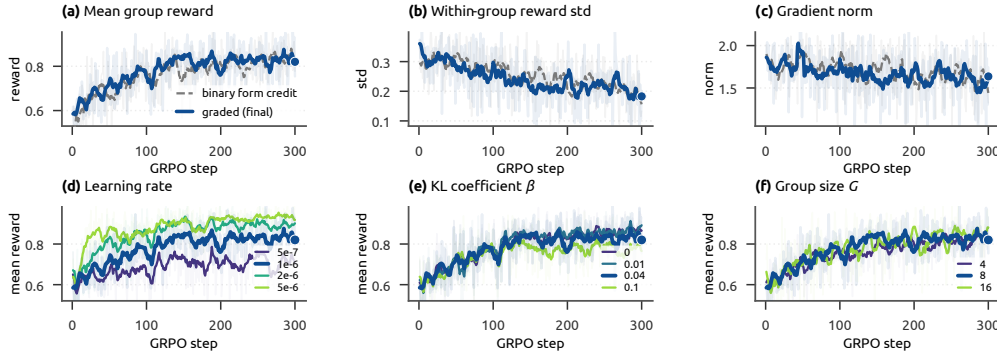


Figure 8: Diagnostic study: **trainer-log quantities behind Figure 5**. (a–c) Mean group reward, within-group reward SD and gradient norm, binary form credit against the final graded run. (d–f) Mean group reward for every hyper-parameter setting of Figure 5(b) on one shared scale; the thick blue curve is the default run ($\text{lr} = 1\text{e-}6$, $\beta = 0.04$, $G = 8$). Curves are smoothed over 9 steps; each axis spans the 2nd–98th percentile of its raw trace, so a few steps clip.

Figure 8 gives the trainer-log quantities behind the diagnostics of Figure 5.

This section records the numbers behind Figure 8. Final KL to the reference falls $0.047 \rightarrow 0.024 \rightarrow 0.010$ as β rises $0.01 \rightarrow 0.04 \rightarrow 0.1$ (the $\beta = 0$ run logs no KL term), and yet the observed mixed rates lie in the narrow range 0.600–0.663 (Table 35): the KL term therefore tracks β as designed, while held-out compliance is flat across the range. Along learning rate, final KL 0.006, 0.024, 0.049, 0.157 and mean group reward 0.713, 0.832, 0.893, 0.936 over 5×10^{-7} to 5×10^{-6} run in the same order as the mixed suite. Across the three parameters of Table 35, structure scores span 0.458 to 0.833 without a monotone trend along any axis; Appendix D.9 adds the reward mixing weight, which reaches 0.325 at $\lambda = 1$.

D.16 ADDITIONAL DIAGNOSTICS

Figure 9 shows the two per-item diagnostics the main text does not: which families fail together, and when each item is first solved during training.

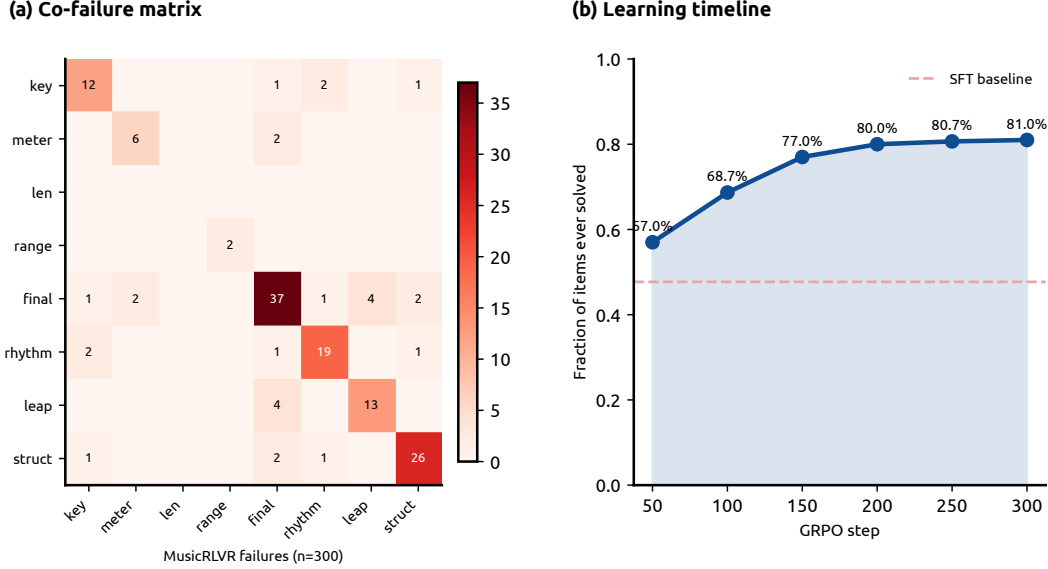


Figure 9: Diagnostic study: per-item diagnostics on the mixed suite ($n = 300$). **(a)** Co-failure matrix of MusicRLVR: diagonal entries are per-family failure counts, off-diagonal entries co-failures, which are rare: key shares at most two with any family. **(b)** CDF of the step at which each item is first solved: 57% by step 50, 81% at least once by step 300.

E ERROR AND STRUCTURE ANALYSIS

E.1 WHICH CONSTRAINT BINDS, BY SPLIT

Figure 10 gives the outcome composition of each split; Table 44 adds the most frequent failing families, using the diagnostic reference evaluation. Figure 11 breaks the same evaluation down by requested form and by how many models solve each item.

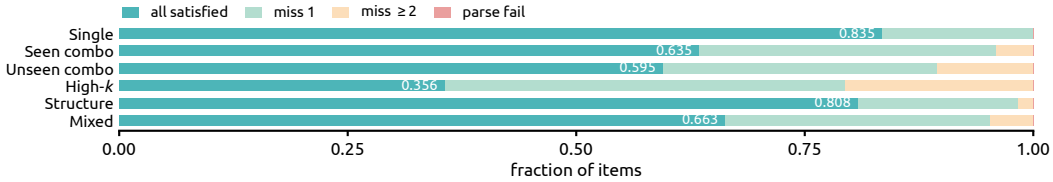


Figure 10: Diagnostic study: **What an unsatisfied item is missing.** Outcome composition for the fixed checkpoint; printed all-satisfied rates match Table 33. Parse failures are zero on all six original splits.

Table 44: Diagnostic study: which constraints fail, by split. “Not all satisfied” is one minus the all-satisfied rate; the next column counts items failing two or more families at once; the last lists the three most frequent failing families as fractions of the split’s items (an item failing two is counted under both). No completion fails the parse gate on any split.

Split	Not all satisfied	Failing ≥ 2 families	Most frequent failing families
Single	0.165	0.000	leap (0.085), final (0.025), key (0.020)
Seen	0.365	0.040	structure (0.125), final (0.120), rhythm (0.065)
Unseen	0.405	0.105	final (0.160), key (0.150), structure (0.090)
High-k	0.644	0.206	final (0.237), rhythm (0.181), key (0.156)
Structure	0.192	0.017	structure (0.133), key (0.042), rhythm (0.033)

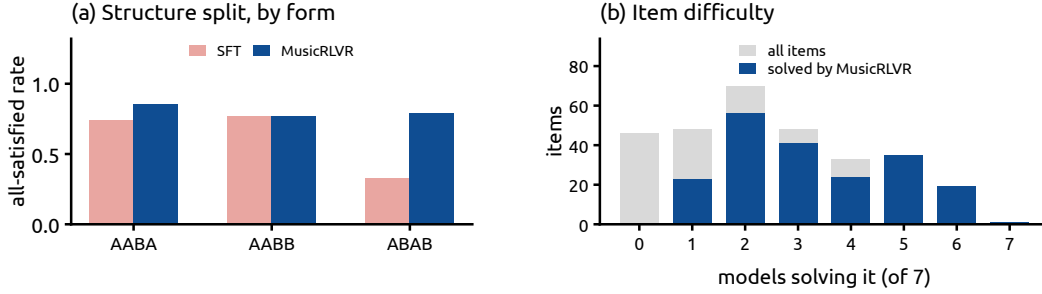


Figure 11: Diagnostic study: **Structure and item-level diagnostics.** (a) Satisfaction by requested form for the fixed checkpoint. (b) Item counts by the number of seven evaluated models solving them under the historical A.

Most unsatisfied items miss one family. The fraction missing multiple families is 0.040 on Seen and 0.105 on Unseen, rising to 0.206 on High- k . The most frequent failing family is `final` on Unseen and High- k . `structure` is among the three most frequent failures on Seen and Unseen; `leap` appears among the top three only on Single. These are frequencies under each split’s own family mixture (Table 38).

E.2 STRUCTURE REWARD VARIANTS

Table 45 compares the fixed default checkpoint with the equal-weight control and the symmetric-credit study. The default and equal-weight rows each use one run; symmetric credit reports the mean and sample SD of its two runs. The default form term weights repetition 0.7 and contrast 0.3; the following controls examine that choice.

Symmetric form credit. The symmetric term uses $0.5\bar{e} + 0.5\bar{d}$, with zero credit for all-identical outputs and credit 1 for outputs passing the binary form check. The outer reward mixture is unchanged. Across two runs, contrast satisfaction is 0.233 ± 0.024 and the all-identical fraction is 0.762 ± 0.018 . The control jointly changes the weights, the contrast factor and the credit for all-identical outputs. Under the default term, an all-identical output earns $s_{\text{structure}} = 0.7$ but has $A = 0$, because it fails contrast. Appendix D.9 separately varies the outer mixing weight.

Reweighting repetition and contrast. The single-variable version moves only the weights, to 0.5/0.5: an all-identical output still banks the repetition half, the binary contrast factor is kept, and the rest of the reward and the training setup are unchanged. Three things follow.

- (1) In the single-run diagnostic comparison, the structure split moves $0.808 \rightarrow 0.675$ and all-identical outputs $15 \rightarrow 32$ of 120, the same direction as the joint control.
- (2) The aggregate symmetric-credit comparison is reported separately in Table 45.
- (3) The equal-weight control scores 0.637 on Mixed, compared with 0.663 for the default.

Appendix D.9 extends this comparison to the outer reward mixture. In the single-run diagnostic study, the two tested λ endpoints yield larger increases in all-identical outputs than the equal-weight form control. The default achieves the highest Structure score and fewest all-identical outputs among these tested weight settings. A term rewarding variation across sections remains a candidate for future evaluation.

Table 45: Diagnostic study: structure decomposition. “Runs” is the number of training runs: default and equal-weight use one each, symmetric credit reports mean $\pm$ sample SD over two. All-identical is a fraction; “all requested” additionally requires the companion constraint, present in 66 of 120 items.

Reward	Runs	Repetition	Contrast	All requested	All-identical
MusicRLVR	1	0.992	0.867	0.808	0.125
Equal weights	1	1.000	0.733	0.675	0.267
Symmetric credit	2	0.996 ± 0.006	0.233 ± 0.024	0.208 ± 0.012	0.762 ± 0.018

E.3 STRUCTURE RESULTS BY FORM TYPE

The structure split contains three form types, in unequal numbers: AABA ($n = 42$), AABB ($n = 35$) and ABAB ($n = 43$). Table 46 decomposes performance on each.

Table 46: Diagnostic study: Structure results by form for the fixed MusicRLVR and SFT checkpoints. Contrast and all-identical are fractions of items of that form.

Model	Form	All-sat.	Contrast	All-identical
SFT	AABA	0.738	0.810	0.190
SFT	AABB	0.771	0.857	0.143
SFT	ABAB	0.326	0.395	0.605
MUSICRLVR	AABA	0.857	0.905	0.071
MUSICRLVR	AABB	0.771	0.829	0.171
MUSICRLVR	ABAB	0.791	0.860	0.140

ABAB is the form supervision handles worst, and by a wide margin: SFT reaches 0.771 on AABB but 0.326 on ABAB, with 26/43 of its ABAB outputs written as four identical units. All three forms require distinct A and B units, so the gap tracks the arrangement rather than the distinctness requirement: ABAB is the only one that interleaves the two units instead of repeating each in place.

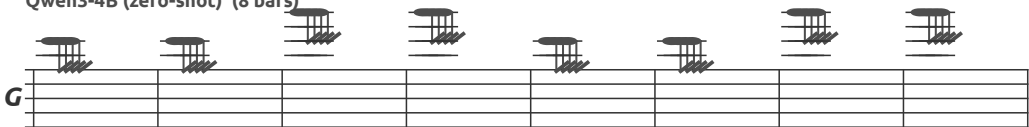
At the fixed checkpoint, ABAB has the largest improvement over SFT among the three forms: its all-satisfied rate rises from 0.326 to 0.791, while the all-identical fraction changes from 26/43 to 0.140. Table 46 reports the rate for every form.

F QUALITATIVE EXAMPLES

Every example below is a benchmark item chosen by a fixed rule (the first item in file order with the stated constraint families and pass/fail pattern), its completions are read verbatim from the frozen evaluation files, and the notation is drawn from music21’s parse of the output: pitch, duration and bar lines. The key and meter checks additionally read the header-derived objects, and the key check analyses the whole score. Marks under each staff are the verifiers’ decisions on that output. Outputs rejected at the gate are shown as a short excerpt together with the gate’s reason. Examples 1–6 compare the zero-shot base model with MusicRLVR (diagnostic run), and Example 2 shows SFT as well; Examples 7–8 compare MusicRLVR with the larger zero-shot models on the same items.

Example 1: unseen-combination split ($k = 2$, held-out pair **key–rhythm**). **Item** uc-0022. **Constraints:** (1) the tune must be in G major; (2) use only sixteenth, eighth, and quarter notes (no dotted notes, no other durations, no rests)

Qwen3-4B (zero-shot) (8 bars)



✗ key G major ✓ rhythm set

MusicRLVR (18 bars; first 16 shown)



✓ key G major ✓ rhythm set

Qwen3-4B (zero-shot) fails **key**; MusicRLVR satisfies every constraint.

Example 2: high- k split ($k = 4$). **Item** hk-0004. **Constraints:** (1) the meter must be 3/4, and every bar must contain exactly the right total duration; (2) the tune must be exactly 8 bars long (no pickup bar); (3) the last note of the tune must be E (any octave); (4) use only eighth and quarter notes (no dotted notes, no other durations, no rests)

Qwen3-4B (zero-shot)

not rendered: parse-error: ABCTokenException

X:1 T:Monophonic Tune M:3/4 L:8 K:C C D E | C D E | E D C | E D C | E D C | E D C | C D E | E D E ...

✗ meter 3/4 ✗ 8 bars ✗ final E ✗ rhythm set

SFT (8 bars)



✓ meter 3/4 ✓ 8 bars ✓ final E ✓ rhythm set

MusicRLVR (8 bars)



✓ meter 3/4 ✓ 8 bars ✓ final E ✓ rhythm set

Qwen3-4B (zero-shot) fails at the parse gate (parse-error: ABCTokenException); SFT satisfies every constraint; MusicRLVR satisfies every constraint.

Example 3: single-constraint split ($k = 1$). Item sg-0000. Constraints: (1) melodic motion must be smooth: no melodic leap larger than an octave

Qwen3-4B (zero-shot)

not rendered: degenerate-or-polyphonic

X:1 ♩ T:Simple Melody ♩ M:4/4 ♩ L:1/4 ♩ K:C

✗ leap ≤ 12

MusicRLVR (16 bars)



✓ leap ≤ 12

Qwen3-4B (zero-shot) fails at the parse gate (degenerate-or-polyphonic); MusicRLVR satisfies every constraint.

Example 4: seen-combination split ($k = 3$). Item sc-0005. Constraints: (1) the last note of the tune must be D (any octave); (2) use only sixteenth, eighth, and quarter notes (no dotted notes, no other durations, no rests); (3) melodic motion must be smooth: no melodic leap larger than a perfect fifth

Qwen3-4B (zero-shot)

not rendered: degenerate-or-polyphonic

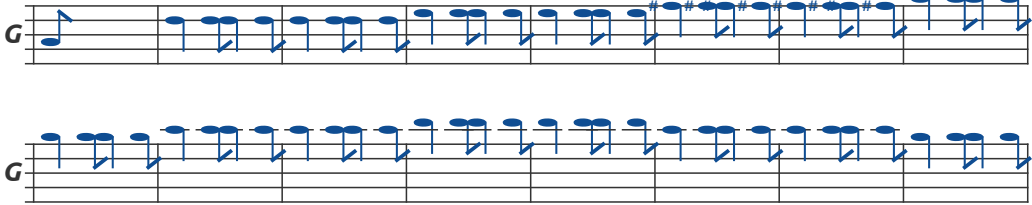
X:1 ♩ T:Simple Tune ♩ M:4/4 ♩ L:1/16 ♩ K:C

✗ final D

✗ rhythm set

✗ leap ≤ 7

MusicRLVR (23 bars; first 16 shown)



✓ final D

✓ rhythm set

✓ leap ≤ 7

Qwen3-4B (zero-shot) fails at the parse gate (degenerate-or-polyphonic); MusicRLVR satisfies every constraint.

Example 5: seen-combination split ($k = 2$). Item sc-0001. Constraints: (1) use only sixteenth, eighth, and quarter notes (no dotted notes, no other durations, no rests); (2) the tune must be exactly 16 bars with an AABB form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ

Qwen3-4B (zero-shot) (8 bars)



✓ rhythm set ✗ form AABB

MusicRLVR (16 bars)



✓ rhythm set ✓ form AABA

Qwen3-4B (zero-shot) fails **structure**; MusicRLVR satisfies every constraint.

Example 6: unseen-combination split ($k = 2$, **held-out pair length-range**). **Item** uc-0001.
Constraints: (1) the tune must be exactly 16 bars long (no pickup bar); (2) every note must lie between G3 and A5 inclusive

Qwen3-4B (zero-shot)

not rendered: degenerate-or-polyphonic
 X:1 ⌕ T:Monophonic Tune ⌕ M:4/4 ⌕ L:1/4 ⌕ K:C

✗ 16 bars ✗ range G3-A5

MusicRLVR (16 bars)



✓ 16 bars ✓ range G3-A5

Qwen3-4B (zero-shot) fails at the parse gate (degenerate-or-polyphonic); MusicRLVR satisfies every constraint.

Example 7: structure split ($k = 1$, **zero-shot models against MusicRLVR**). **Item** st-0023.
Constraints: (1) the tune must be exactly 16 bars with an AABA form in 4-bar units: labeled units with the same letter must be note-for-note identical, and units with different letters must differ

Qwen3-32B (zero-shot) (15 bars)



✗ form AABA

Llama-3.1-70B (zero-shot) (32 bars; first 16 shown)



✗ form AABA

MiniMax-M3 (zero-shot)

not rendered: repeat-signs-forbidden

X: 1 ♩ T: AABA Tune ♩ M: 4/4 ♩ L: 1/4 ♩ K: G ♩ |: G2 B2 | d4 B2 A2 | G4 D2 | E4 G2 A2 | ♩ B2 d2 | g4 d2 B2 | A...

✗ form AABA

MusicRLVR (16 bars)



✓ form AABA

Qwen3-32B (zero-shot) fails structure; Llama-3.1-70B (zero-shot) fails structure; MiniMax-M3 (zero-shot) fails at the parse gate (repeat-signs-forbidden); MusicRLVR satisfies every constraint.

Example 8: high- k split ($k = 4$, zero-shot models against MusicRLVR). Item hk-0278. Constraints: (1) the tune must be in C major; (2) every note must lie between D4 and D6 inclusive; (3) the last note of the tune must be G (any octave); (4) melodic motion must be smooth: no melodic leap larger than a perfect fifth



Qwen3-32B (zero-shot) fails *range*, *final*; Llama-3.1-70B (zero-shot) fails *range*; MusicRLVR satisfies every constraint.

G BROADER IMPACT

This section expands the ethics statement. The method makes music-generation models follow explicit specifications, which helps wherever a score must meet a specification: music education, game audio, accessibility tools. We do not foresee significant negative societal impacts specific to this work, because the experiments concern short monophonic melodies in ABC notation under explicit constraints and evaluate neither production use nor perceptual quality. That last point is also where the method can be misread: a high all-satisfied rate says the requested properties hold, not that the tune is good; an output can satisfy every stated constraint while repeating one short motif (Figure 3), and perceptual quality is not measured here. All supervised targets derive from the publicly available IrishMAN corpus of traditional Irish music; the reinforcement prompts are generated synthetically. The benchmark and verifiers will be released to support reproducibility and further research.